3D Shape Matching and Registration

– A Probabilistic Perspective

Pengdong Xiao

THE AUSTRALIAN NATIONAL UNIVERSITY

A thesis submitted for the degree of Doctor of Philosophy of
The Australian National University

March 2010

Declaration

This thesis is submitted to the School of Engineering, ANU College of Engineering and Computer Science, The Australian National University, in fulfilment of the requirements for the degree of Doctor of Philosophy.

This thesis is entirely my own work, except where otherwise stated, describes my own research. It contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning.

Pengdong Xiao

26th March 2010

Acknowledgements

I wish to thank Dr Nick Barnes, my thesis advisor, for all the guidance through my Ph.D. program at ANU. I wish to thank Dr Tiberio Caetano for guiding me on Markov random fields. I wish also to thank Dr Paulette Lieby for research and technical support.

I would like to thank the Open Access Series of Imaging Studies (OASIS) for making the MRI data sets publicly available to use. I also thank the Personality and Total Health (PATH) Through Life Project research team at the Centre for Mental Health Research, ANU, for providing the MRI scans and the segmented data sets.

I am grateful to IT people in Canberra Research Laboratory, National ICT Australia (NICTA) for the consistent IT support.

I would like to thank Luping Zhou for valuable discussions.

This work was generously supported by NICTA.

List of Publications

A number of publications have directly resulted from this research. These are

1. Pengdong Xiao, Nick Barnes, Tiberio Caetano, and Paulette Lieby. An MRF and Gaussian Curvature Based Shape Representation for Shape Matching. In *IEEE CVPR Workshop on Beyond Multiview Geometry: Robust Estimation and Organization of Shapes from Multiple Cues*, Minneapolis, USA, June 2007.

2. Pengdong Xiao. Randomized Algorithms to Minimize the Energy Function for 3D Shape Matching. In *The Sixth IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA 2009)*, Innsbruck, Austria, February 2009.

3. Pengdong Xiao, Nick Barnes, Paulette Lieby, and Tiberio Caetano. Applying Sum and Max Product Algorithms of Belief Propagation to 3D Shape Matching and Registration. In *Digital Image Computing: Techniques and Applications (DICTA 2009)*, Melbourne, Australia, December 2009.

Abstract

Dense correspondence is a key area in computer vision and medical image analysis. It has applications in registration and shape analysis. In this thesis, we develop a technique to recover dense correspondences between the surfaces of neuroanatomical objects over heterogeneous populations of individuals. We recover dense correspondences based on 3D shape matching.

In this thesis, the 3D shape matching problem is formulated under the framework of Markov Random Fields (MRFs). We represent the surfaces of neuroanatomical objects as genus zero voxel-based meshes. The surface meshes are projected into a Markov random field space. The projection carries both geometric and topological information in terms of Gaussian curvature and mesh neighbourhood from the original space to the random field space. Gaussian curvature is projected to the nodes of the MRF, and the mesh neighbourhood structure is projected to the edges. 3D shape matching between two surface meshes is then performed by solving an energy function minimisation problem formulated with MRFs. The outcome of the 3D shape matching is dense point-to-point correspondences.

However, the minimisation of the energy function is NP hard. In this thesis, we use belief propagation to perform the probabilistic inference for 3D shape matching. A sparse update loopy belief propagation algorithm adapted to the 3D shape matching is proposed to obtain an approximate global solution for the 3D shape matching problem. The sparse update loopy belief propagation algorithm demonstrates significant efficiency gain compared to standard belief propagation. The computational complexity and convergence property analysis for the sparse update loopy belief propagation algorithm are also conducted in the thesis. We also investigate randomised algorithms to minimise the energy function.

In order to enhance the shape matching rate and increase the inlier support set, we propose a novel clamping technique. The clamping technique is realized by combining the loopy belief propagation message updating rule with the feedback from 3D rigid body registration. By

using this clamping technique, the correct shape matching rate is increased significantly.

Finally, we investigate 3D shape registration techniques based on the 3D shape matching result. Based on the point-to-point dense correspondences obtained from the 3D shape matching, a three-point based transformation estimation technique is combined with the RANdom SAmple Consensus (RANSAC) algorithm to obtain the inlier support set. The global registration approach is purely dependent on point-wise correspondences between two meshed surfaces. It has the advantage that the need for orientation initialisation is eliminated and that it handles all shapes of spherical topology. The comparison of our MRF based 3D registration approach with a state-of-the-art registration algorithm, the first order ellipsoid template, is conducted in the experiments. These show dense correspondence for pairs of hippocampi from two different data sets, each of around 20 60+ year old healthy individuals.

Contents

Notation

- M, M' – index set for a graph, $M = \{1, ..., m\}$ and $M' = \{1, ..., n\}$;
- i – a node in a graph;
- x_i – state for node i;
- X_i – random variable at a single node i;
- S – state space for every single node $i \in M$;
- x – one instance of configuration $x = (x_i)_{i \in M}$;
- X – random field, $X = \prod_{i \in M} X_i$;
- Ω – configuration space of random field X, $\Omega = S^m$;
- $N(i)$ – neighbourhood of node i;
- N_M – neighbourhood system of a graph, $N_M = \{N(i)\}_{i \in M}$
- (M, N_M) – random field graph;
- C – set of cliques in graph (M, N_M);
- K – $K(u)$ denotes the Gaussian curvature of u, u a node of M or M';
- V_C – potential for clique C;
- P – probability;
- g – Gaussian kernel;
- $m_{ij}(x_j)$ – the message from node i to node j about what state node j should be in;
- $b_i(x_i)$ – the belief at a node i about the state x_i;
- H – energy function;
- H_1 – energy function based on singletons;
- H_2 – energy function based on 2-element cliques;
- h_1 – potential function defined on singletons;
- h_2 – potential function defined on 2-element cliques;
- w – balancing factor between internal and external energy;

- d_m – Minkowski distance;
- d_p – pairwise distance;
- $\phi_i(x_i)$ – compatibility function based on curvature difference;
- $\psi_{ij}(x_i, x_j)$ – compatibility function based on pairwise distance;

Chapter 1

Introduction

The basic goal of this thesis is to present methods for matching structurally different 3D objects and recovering dense point-to-point correspondence between the surfaces of two shapes. Investigation of this problem is motivated by Alzheimer's Disease (AD) and other general neuro-degenerative conditions which lead to volume loss and shape change of neuroanatomical structures, and need good techniques for early diagnosis. Alzheimer's is a progressive disease in which atrophy occurs in major components of the brain over time [1].

AD is linked to changes in shape of brain structure. It has been shown [39] that in patients with AD, the areas significantly affected by the disease include the corpus callosum, the hippocampus, the amygdala, and the thalamus. A human brain image is shown in Figure 1.1. In this thesis, we use the hippocampus as a case study in our generic dense correspondence matching method. A segmented hippocampus is shown in Figure 1.2. The hippocampus is an area in the forebrain that regulates emotion and memory and affects spatial navigation. Each person has two hippocampi. Changes in hippocampal shape and structure have been linked to AD and other neuro-degenerative diseases. It is understood that hippocampal volume loss and shape change occurs well before actual diagnosis of AD. Histological studies have shown that the hippocampus is particularly vulnerable to Alzheimer's disease pathology and is already considerably damaged at the time clinical symptoms first appear [23]. Therefore, the analysis of hippocampus shape change may facilitate early diagnosis of AD, and also facilitate new treatments and early intervention. Detecting disease before clinical symptoms appear is critical when there are treatments that can prevent more extensive damage and halt the progression of the disease [26].

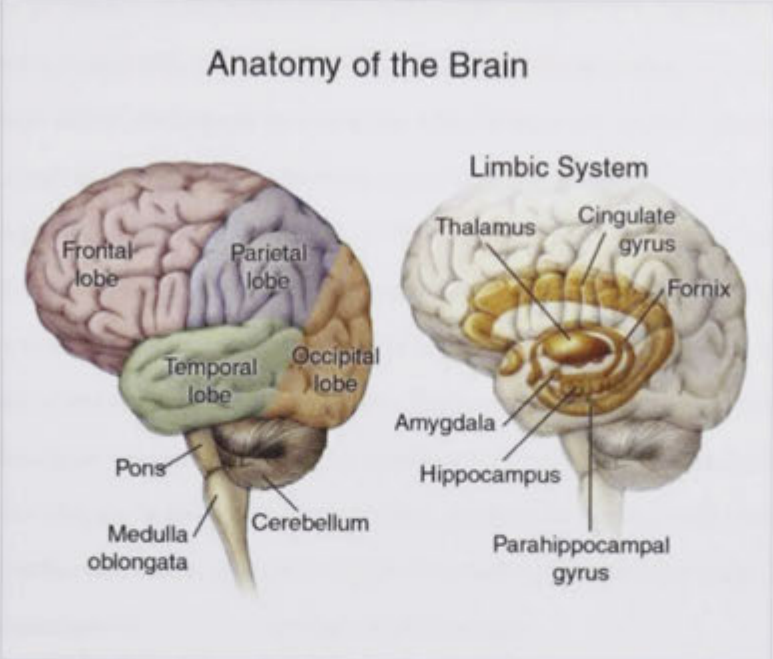


Figure 1.1: Human brain. The figure is taken from http://www.pakmed.net/academic/age/alz/alz028.htm/.

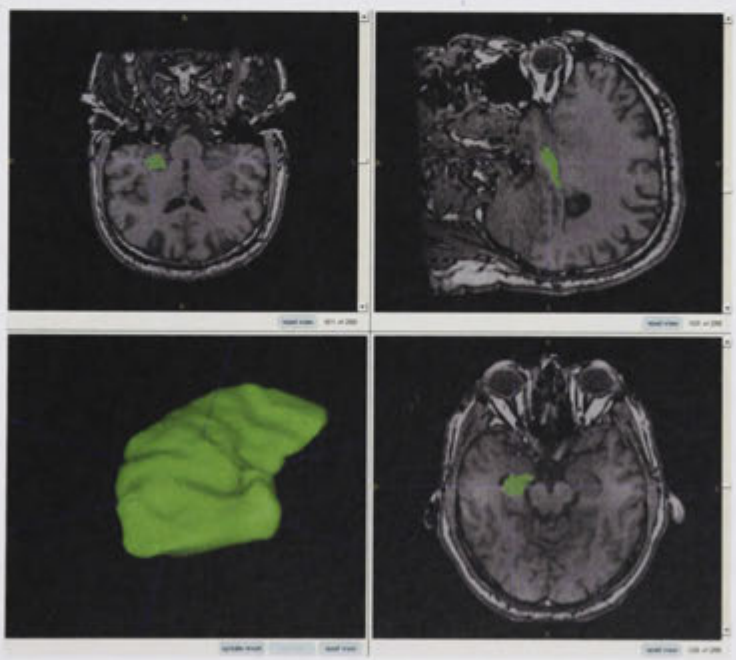


Figure 1.2: A segmented hippocampus.

The shape difference of the hippocampus induced by Alzheimer's can facilitate the classification between control and AD as suggested in [33, 161]. Shape analysis can be used both to detect AD early and to understand its evolution. Originally, most studies focused on volume [49, 14] and shape histograms [88]. However, shape analysis enables the study of AD in terms of shape change with respect to disease [33, 145]. Although volume and shape histograms can give some information for classification, point-to-point or region-to-region correspondence offers greater precision. Obtaining it is difficult due to the large variability between shapes, which becomes more severe due to the ageing process. Region-based shape analysis may help refine our understanding of the impact of AD. An excellent way to achieve this is dense correspondences between shapes. In this thesis, we contribute methods for finding such correspondences between the surfaces of neuroanatomical objects over heterogeneous populations. We assume that the neuroanatomical structures have spherical topology.

Although our work to recover dense correspondences between two 3D shapes is motivated by the biomedical application to provide a method to perform detailed point-to-point based comparison between two 3D neuroanatomical objects, the technique is general and as such can be used for matching any kind of 3D shapes which have spherical topology. Other potential applications of our 3D shape matching technique include object recognition, indexing and retrieval of 3D objects from databases, registration, and classification.

1.1 Approach

We explore the application of Markov random fields (MRFs) techniques to shape matching. The nature of our matching problem motivates us to use Markov random fields to solve our problem. The purpose of our shape matching is to find correspondences between two 3D shapes based on features which exhibit randomness in nature. Since our 3D shapes are represented as meshed surfaces, the natural way to model the matching problem would be to use Markov random fields which can capture the randomness and spatial mesh structure of the 3D shapes. More importantly, by defining cliques of singletons and neighbouring pairs, according to Markov property that will be described in Section 3.1, the conditional probability of the

matching correspondences, which are equivalent to the state on a node of a Markov random field, is determined based on the feature difference on singletons and pairwise distance on neighbours. The cliques of singletons and neighbouring pairs can capture rich structural information of 3D neuroanatomical objects while performing matching. The Markov assumption is important here. Firstly, it allows us to use cliques to capture the structural information. Secondly, it helps us to transform the correspondence matching problem into a conditional probability estimation problem. Therefore, Markov random fields are the most suitable technique to solve our matching problem. As mentioned, our 3D shape matching is carried out between two surfaces represented as meshes. The outcome of the 3D shape matching is point based dense correspondences. The surface mesh is projected into a Markov random field space. The projection preserves both geometric and topological information of the original shape. During the projection, the geometric information which is measured as Gaussian curvature is preserved at the nodes of the MRF, and the topological information is preserved as the spatial pairwise distance at the edges of the MRF. Matching between two surfaces is performed based on the projection.

We adopt a similar approach to the one introduced in [29]. Let M and M' be two meshed surfaces, and X be an MRF. We perform a one-to-one mapping $M \rightarrow X$ which preserves Gaussian curvature at nodes and mesh topological structure at edges in X. The correspondence matching problem between M and M' is then an energy function minimisation problem in X, and a point correspondence between M and M' is a realization of the MRF X in M'.

Under the MRF framework, 3D shape matching based on meshed surfaces can be formulated as an energy function minimisation problem. However, to solve such a global optimisation problem is NP-hard. So research mainly focuses on approximate algorithms. One of the best known is belief propagation (BP), which has shown success in early vision and many other practical applications. In this thesis, we investigate the application of both sum and max product algorithms of belief propagation to 3D shape matching.

1.2 Contributions

We develop an approach for recovering dense correspondences between the surfaces of neuroanatomical objects over heterogeneous populations. The recovery of dense correspondences is performed by 3D shape matching.

We further formulate the 3D shape matching technique under the framework of Markov random fields. The 3D shape matching is formulated as an energy function minimisation problem defined based on Gaussian curvature and spatial pairwise distance. The NP hard minimisation problem of the energy function is carried out by using loopy belief propagation.

We propose a modified sparse loopy belief propagation algorithm adapted to 3D shape matching to obtain an approximate global solution for a 3D shape matching problem, which shows significant efficiency gain compared to standard BP. We also perform the computational complexity and convergence property analysis for the modified algorithms.

In order to enhance the shape matching rate and increase the inlier support set, we propose a clamping technique. The clamping technique is realized by integrating the loopy belief propagation message updating rule and the feedback from the 3D rigid body registration. The correct shape matching rate can be significantly increased.

We also explore the use of randomised techniques to solve the energy function minimisation problem under Markov random fields for 3D shape matching between two meshed surfaces embedded in $\mathbb{R}^3$. We investigate the use of simulated annealing based on the Gibbs sampler and Metropolis sampler and a Las Vegas type algorithm to minimise the energy function.

Based on the 3D shape matching result, we further investigate 3D shape registration techniques. In this thesis, a 3D shape registration technique based on 3D shape matching is presented. The advantage of the proposed 3D rigid body registration method based on 3D shape matching is the elimination of the need for orientation initialisation. The registration is global and it is purely dependent on point-wise correspondence between two meshed surfaces from 3D shape matching. From the 3D matching results between two meshed surfaces, the three-point based transformation estimation algorithms and RANSAC are used to eliminate the outliers, and get the inlier support. From the inlier support, optimal 3D rigid body transformation

estimation techniques are used to finally register two 3D shapes.

The major contributions of the thesis are summarised as follows:

- The definition and empirical verification of an efficient technique for recovering dense correspondences between the surfaces of neuroanatomical objects over heterogeneous populations.
- The formulation of the 3D shape matching problem between objects within a Markov random field framework.
- A novel sparse update loopy belief propagation algorithm to perform the probabilistic inference in this problem, which is much faster than the standard BP.
- A novel clamping technique by integrating belief propagation and an external measure of the quality of match to enhance the shape matching rate and increase the inlier support set.
- The empirical evaluation of randomised algorithms of simulated annealing based on Gibbs sampler and Metropolis sampler to minimise the energy function for 3D shape matching.
- The demonstration of 3D rigid body registration based on 3D shape matching results. The registration is global. It purely depends on matching results, and no initial alignment is needed. The 3D rigid body registration results based on 3D shape matching are comparable to a state-of-the-art registration algorithm.

1.3 Thesis Structure

The thesis is organised as follows.

Chapter 2 presents the literature review, including shape representation and shape matching, dense correspondence, rigid and non-rigid registration.

Chapter 3 provides the background knowledge on Markov random fields, belief propagation, Gaussian curvature and scale space theory, and 3D rigid body transformation estimation

based on three pairs of corresponding points.

Chapter 4 describes a theoretical model for 3D shape matching based on Markov random field theory and techniques.

Chapter 5 gives a novel modified belief propagation algorithm – sparse update belief propagation, including its computational complexity and convergence property analyses. A novel clamping technique is also presented in this chapter, which significantly increases the correspondence matching rate.

Chapter 6 presents an empirical analysis of the randomised algorithms to minimise the energy function. In order to show alternative approaches to minimising the energy function besides belief propagation, we also investigate simulated annealing algorithms for the minimisation problem. Both Gibbs sampler and Metropolis sampler based simulated annealing and a Las Vegas type algorithm to minimise the energy function are presented and evaluated in this chapter.

Chapter 7 presents the 3D rigid body registration method. The techniques of using a three point based transformation estimation method combining with RANSAC to eliminate the correspondence outliers, and obtain the inlier support set are illustrated. The optimal transformation estimation approaches based on eigenvalue decomposition, singular value decomposition, and quaternions are also presented in this chapter.

Chapter 8 gives the clamping experimental results and 3D rigid body registration results. The experiments with both synthetic and real data are presented in the chapter. The real data include two large databases with large real data sets.

Chapter 9 concludes the thesis and discusses the future work.

Chapter 2

Literature Review

3D shape matching is to determine the quantitative and qualitative relationship between two 3D shapes. Sometimes shape matching is formulated at the level of dense point-to-point based correspondence matching between two 3D objects. Some methods enforce topological consistency between the point-to-point matches, and some do not. Dense correspondence can be used as a basis for rigid and nonrigid registration.

2.1 3D Shape Representation and Matching

3D shape matching is a key area in computer vision and medical image analysis. It deals with the issue of how to measure the similarity between two 3D shapes and establish correspondences between the shapes. 3D shape matching has many applications such as shape registration, 3D object recognition and retrieval, and classification. In this thesis, we will mainly investigate how 3D shape matching can be used to generate dense correspondence between two 3D objects.

Currently there is an increasing demand to recognise and retrieve 3D shapes from domain-specific databases efficiently and accurately, such as Computer-Aided Design (CAD) [64, 76], 3D object retrieval [157, 158], molecular biology [68], and chemistry [27]. 3D shape matching is one of the key issues to solve the recognition and retrieval problem over large databases.

There is a tightly coupled issue with shape matching, which must be addressed first, called shape representation or shape description. How a shape is represented or described has a direct influence on how shape matching is performed. Invariance, uniqueness, and stability are the desired properties for an effective and efficient representation [108].

2D shape representation and matching has been relatively better investigated than 3D, and remains an active research area. The early work includes Fischler and Elschlager's mass-spring model to find a visual object in an actual photograph by combining a descriptive scheme with a decision metric [40], and Fourier descriptors [156, 116]. It has been addressed in several other ways, including wavelet transform [31, 138, 123], level sets [142], curvature scale space [109, 105, 36, 141, 159], and shape contexts [16]. More recently an approach based on discrete Fourier coefficients of the boundary values is proposed in [79]. The method is named Shape Signature Harmonic Embedding, and its principle is to solve a harmonic function embedded in a circular disk with a discrete Poisson kernel and Dirichlet boundary condition.

Some approaches to 3D shape matching are the natural generalisation from 2D, such as spherical harmonics [125, 70], 3D shape contexts [75], and local surface curvatures [34]. There are other 3D shape matching techniques developed independently from 2D, such as spin images [66] and skeleton shape descriptor [135]. Tangelder et al. provide a complete survey on 3D shape matching in [136]. In the following, we introduce some popular 3D shape representation and matching methods.

2.1.1 Spherical Harmonics

The spherical harmonics (SPHARM) description is a parametric surface description that only represents objects which have spherical topology [24]. The surface is parameterised by using spherical harmonics as basis functions. Kelemen et al. [71] use SPHARM to represent 3D neuroanatomical objects, and have shown that SPHARM can be used to express shape deformations [71]. In [133] Styner and Gerig propose a shape representation approach that compute a stable three-dimensional medial model in the presence of shape variability based on M-rep [118] and SPHARM.

In [24], the surface is parametrised by defining a continuous, one-to-one mapping from the surface of the original object to the surface of a unit sphere. The parametrisation is formulated as a constrained optimisation problem by Brechbuhler [24]. However, the basic parametrisation algorithm based on the constrained optimisation problem becomes unstable when object

meshes have more than a thousand vertices [121]. Quicken et al. improve the basic algorithm by handling the situation when object meshes consist of large number of vertices [121]. The improvement is realized by using multi-resolution meshes through a hierarchical optimisation approach. The improved parametrisation procedure can handle cases with several thousand vertices. From the parametrisation as suggested by Brechbuhler, we obtain three explicit functions defining the object surface as follows [24]:

$$\mathbf{x}(\theta,\phi) = \begin{pmatrix} x(\theta,\phi) \\ y(\theta,\phi) \\ z(\theta,\phi) \end{pmatrix},$$

where $\theta = 0 \cdots \pi, \phi = 0 \cdots 2\pi$.

Let Y_l^m denote the spherical harmonic function of degree l and order m

$$Y_l^m(\theta,\phi) = \sqrt{\frac{2l+1}{4}\frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta) e^{im\phi},$$

$$Y_l^{-m}(\theta,\phi) = (-1)^m Y_l^{m*}(\theta,\phi)$$

where Y_l^{m*} is the complex conjugate of Y_l^m, and P_l^m is an associated Legendre function [24]. A spherical harmonic function of degree l is written as a homogeneous polynomial of degree l in $u = \sin\theta\cos\phi$, $v = \sin\theta\sin\phi$ and $w = \cos\theta$.

The surface is expressed as the following spherical harmonic series,

$$\mathbf{x}(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \mathbf{c}_l^m Y_l^m(\theta,\phi),$$

where the coefficients in this series are three-dimensional vectors,

$$\mathbf{c}_l^m = \begin{pmatrix} {c_x}_l^m \\ {c_y}_l^m \\ {c_z}_l^m \end{pmatrix}.$$

The coefficients of the spherical harmonic functions of different degrees provide a measure of the spatial frequency constituents that comprise the structure. The sums of the series are truncated by limiting l. Truncating the series at different degrees results in object representations at different levels of detail. As higher frequency components are included, more detailed features of the object appear [24].

In [69] Kazhdan described a new 3D model matching and indexing algorithm that uses SPHARM to compute discriminating similarity measures without requiring repair of model degeneracies or alignment of orientation. A 3D object is represented as a polygonal model. All 3D objects are assumed to be genus zero objects. The steps for computing the spherical harmonics shape representation is as follows [69]:

1. Given a surface model, its polygons are rasterized into a voxel grid of a certain size. The model is aligned so that its centre of mass is at the centre of the grid.

2. The voxel grid is decomposed into spherical functions by confining the voxel grid to spheres with radii of specified sizes.

3. Each of these spherical functions is decomposed as a sum of its first 16 harmonic components, analogous to a Fourier decomposition into different frequencies.

4. By using the fact that rotations do not change the norm of the harmonic components, the signature of each spherical function is defined as a list of these 16 norms.

5. Finally, the shape descriptor for the 3D model can be obtained by combining these different signatures.

The obtained shape descriptor is rotation invariant [70]. The comparison of two shape representations can be performed by simply computing the Euclidean distance between them. However, it is not obvious how to use the spherical harmonics to recover the dense correspondence between two objects due to its parametric form of shape representation of a surface.

2.1.2 Shape Contexts

For a given point P on a 2D shape with N sampled points, the histogram of the relative coordinates of the remaining N-1 points is defined to be the 2D shape context of P. Usually a log-polar coordinate system is used to generate the histogram. In [75] Kortgen et al. introduce a 3D approach by extending 2D shape contexts that can be used for measuring 3D shape similarity as a fast, intuitive and powerful similarity model for 3D objects. The 3D shape context of a 3D shape is a set of N histograms corresponding to N points sampled from the shape boundary.

The shape context at a point captures the distribution over relative positions of other shape points and thus summarises global shape in a rich, local descriptor [75]. In this way, shape similarity can be formulated as a shape distribution represented as histograms [113]. The shape context describes the coarse distribution of the rest of the shape with respect to a given point on the shape. After representing shape similarity as a shape distribution, the shape matching problem can be reduced to the comparison of probability distribution, which is simpler than traditional shape matching methods that require pose registration, feature correspondence, or model fitting.

In addition, the shape contexts greatly simplify the recovery of correspondences between points of two given shapes [75]. Moreover, the shape context approach leads to a robust scope for measuring shape similarity, once shapes are aligned. One limitation of the shape context is that it is only appropriate for category-level object recognition. This is mainly because the shape context describes just the coarse distribution of the rest of the shape for a given point on the shape. Although the shape context can be used to recover correspondences between points of two given shapes, there is no literature combining shape contexts with a method enforcing topological consistency (eg MRFs) when recovering the dense correspondences.

2.1.3 Spin Image

A spin image is a 2D histogram that encodes the density of mesh vertices projected onto an object-centred space [66]. The bins are the coordinate index in a plane. The 2D histogram is

formed by spinning a plane around a normal vector on the surface of the object and counting the points that fall inside bins in the plane. A spin image is a data level shape descriptor that is used to match surfaces represented as surface meshes [66]. An oriented point is a 3D point with surface normal. A spin image is created for an oriented point at a vertex in the surface mesh as follows:

1. Given an oriented point on the surface and its neighbourhood of a certain size, the normal vector and the tangent plane can be calculated at that point. A 2D basis local coordinate system is established based on the normal vector and the tangent plane. From the coordinate system, an object-centred coordinate (α, β) is computed for each vertex, where α is the distance to the normal, and β is the distance to the tangent plane of the oriented point corresponding to a vertex.

2. A 2D accumulator indexed by (α, β) is created and the accumulator is incremented for each vertex of an object surface.

3. The final 2D histogram is generated.

The spin image shape representation uses information from the entire surface of the object since all the points on the surface are projected to the bins of the spin image [65]. Since spin images describe the relative position of points on a rigid object with respect to a particular point on that same object, spin images are independent of rigid transformation applied to the object. Therefore, the spin image representation is an object-centred shape descriptor [65]. The limitation of the spin image descriptor is that it requires the surface of an object to be uniformly sampled. During matching, the spin images between model and scene are compared by computing correlation coefficients to produce the point correspondences between model and scene, which are used to compute the transformations. However, the spin image is a rather local shape descriptor that focuses on local features. It is difficult to use the spin image to perform 3D shape matching for neuroanatomical objects such as the hippocampus since normally neuroanatomical objects do not have obvious locally distinguishable features. As shown in [6], the spin image has been used to recover dense correspondence for articulated

objects by combining with a Markov network. We have a detailed discussion on this in 2.2.2.

2.1.4 Skeleton Shape Descriptor

Skeleton shape descriptors [135] encode geometric and topological information in the form of a skeletal graph and uses graph matching techniques to match the skeletons and to compare them. Matching two graphs is formulated as a largest isomorphic subgraph problem. At each node in the graph, a structural signature is defined, which characterises the node's underlying subgraph structure. This signature is a low-dimensional vector whose components are based on the eigenvalues of the subgraph's (0,1) adjacency matrix. Since the contextual graph structure is encoded in a node's signature vector, the graph matching is reformulated as a problem of finding the maximum cardinality, minimum weight matching in a bipartite graph. In such a formulation, there is an edge between each node in one graph and each node in the other, whose edge weight represents the distance between the two nodes' structural signature vectors. Solving for maximum cardinality, minimum weight matching involves choosing a subset of the edges in the bipartite graph which provides a one-to-one mapping, whose sum edge weights is small, and whose cardinality is high.

Skeleton based shape matching approach is to use a matching methodology based upon a skeleton of a 3D shape. The skeleton used in this context is a graph-like representation of a 3D object. The skeleton graph accommodates the topological information about the 3D object in terms of the graph and local shape descriptors, which are held at each node in the graph [135]. These local shape descriptors contain information which is used during the matching process such as mean, radius, degree of freedom about the joint. The skeleton shape descriptor is obtained through a 3D skeletonization process, including volumetric thinning, clustering, and generation of the skeletal graph. The skeleton shape descriptor also supports part matching and articulated object matching. It supports part matching because it is a local shape measure descriptor. Since the skeleton topology does not change during articulated motion, articulated object matching is also supported by the skeleton shape descriptor. However, it is not natural to use the skeleton shape descriptor to recover dense correspondence between the surfaces of

two objects.

Siddiqi et al propose a skeleton based graph named shock graphs for 2D shape matching in [130], which are graphs constructed by connecting shocks derived from a curve evolution process. A shock is a singularity during the curve evolution process. Pelillo et al investigate the problem of how to match two shock trees by constructing association graphs in [115]. An association graph is derived from two rooted trees based on the connectivity and the distance matrix.

Leymarie and Kimia extend shock graphs to 3D and propose a hierarchically structured graph, shock scaffolds, to represent 3D shapes [82]. The starting point of the representation is the medial axis (MA) which has been used to represent shapes in 2D. The nodes of a shock scaffold correspond to the singularities of the shock flows along the MA, and the links represent curve segments [81].

Siddiqi et al develop a method for detecting singularities or shocks of an evolution process in [128] and Chapter 4 of [129]. The advantage of the algorithm is its low computational complexity. Borgefors et al describe two skeletonization algorithms for 2D images and 3D images respectively in Chapter 5 of [129]. The limitation of the algorithm is that it assumes the data is isotropic.

2.1.5 Local Surface Curvature

The similarity measure between two shapes can be performed by comparing the corresponding curvature distributions generated from the two surface meshes [127]. The comparison of two curvature distributions can be carried out by using a 3D shape similarity metric defined based on L_p distance. L_2 distance is used in all experiments in [127] for the ease of computation.

Among the shape descriptors, curvature scale space (CSS) is a popular representation, and has been selected as contour-based shape descriptor for MPEG-7 for 2D shape matching. A CSS representation is a multi-scale organisation of the invariant geometric features (curvature zero-crossing points and/or extrema) of a planar curve. Curvature zero-crossings are points where the curvature function of a planar curve changes sign from positive to negative or from

negative to positive. Although CSS has shown superior performance over other descriptors [108], for example in contour based image indexing and retrieval [106], it is mainly used for shape description for still image and video. As one example, it has been successfully used as a feature vector to retrieve images from multimedia databases by calculating the maxima of curvature zero-crossing contours, and then to carry out the matching by using the similarity measure of the maxima [107].

However, to date, using curvature to represent shape for 3D models has not been investigated as much as for image and video. Though some pioneering work on 3D shape representation and matching with curvature has been done [89], there is little work on how to represent and match shapes for 3D surfaces while preserving both geometric and topological properties. In [89], the representation and matching of surfaces with Gaussian curvature is presented. A surface shape representation called the unit normal and shape descriptors list array (UNSDLA) is proposed. A UNSDLA is represented as a 2D edge-node graph and can be stored as a multi-link list in computer implementation. Each node represents a connected surface patch whose unit normals fall within a tessellated cell on the unit sphere. A node stores the average Gaussian curvatures of the surface patch. The matching algorithm requires a tessellation of the unit sphere. The tessellation of the unit sphere can be represented in a matrix form with one matrix for the upper hemisphere and one matrix for the lower hemisphere. During surface matching, the two mapping matrices are used to establish the correspondence between a model surface and a sensed surface. When matching a sensed surface to an object model surface, every unit normal of the sensed surface is mapped to a tessellated cell on the unit sphere. The corresponding node in the UNSDLA of the object model surface is found through the mapping matrices. A matching error between the corresponding shape descriptors and vectors is then computed. It is currently an active research area how to effectively represent shape with curvature in three dimensional space and use the representation for shape matching and statistical analysis [44, 37].

Although the local surface curvature can be used to measure the curvature at a node of a surface mesh, an approach on how to combine a global framework such as Markov random

fields with the local surface curvature is lacking in the literature.

In this thesis, we will investigate using curvature as a descriptor to represent 3D shapes and perform 3D shape matching to recover dense correspondence. We will use curvature in particular scales to carry out the 3D shape matching.

2.2 Dense Correspondence

In this thesis, we aim to find point-to-point dense correspondence between two objects. The majority of 3D shape matching methods do not generate dense correspondences. In the following, we review some literature where dense correspondences are generated while enforcing topological consistency. This is most relevant to the thesis's contribution, the recovery of dense point-to-point correspondences.

2.2.1 Spectral Graph Theory Approach

Mateus et al. solve the dense correspondence matching problem for articulated shapes by using spectral graph theory and the expectation-maximisation (EM) algorithm [99, 98, 97].

The shape is represented as a dense set of voxels. In [99], Mateus et al. conduct the dense correspondence matching by using Laplacian embedding with probabilistic point matching. The articulated objects are described by the Laplacian matrix, which can capture both geometric and topological properties of a graph [99]. The Laplacian matrix is constructed from a weighted adjacency matrix whose entries are obtained based on the voxel connectivity and the local shortest-path distance between points of an object. After graph Laplacian embedding, the correspondence matching problem is cast into an unsupervised point registration problem, which is solved by using EM to maximise the conditional expectation taken over the joint log-likelihood of observations and assignments [99]. The output of the unsupervised point registration is the dense point-to-point matching result.

In addition, Mateus et al. formulated articulated shape matching with locally linear embedding (LLE) to perform dense correspondence matching in the embedding space [98, 97]. LLE is an unsupervised learning algorithm that computes the embedding of a set of input

points while preserving the distances between each point and its neighbours of certain size [98]. Similarly, an EM algorithm is used to evaluate the probabilities of each possible point pair, and maximise the expectation of the logarithm of the joint probability of the observations and their correspondences. Finally, the one-to-one assignment is obtained from the assignment probabilities calculated from the EM algorithm.

The spectral graph theory based approach uses an EM algorithm to maximise the conditional expectation. An initial orthogonal transformation R and a covariance matrix Σ need to be provided for EM initialisation. This requires a good initialisation for the EM algorithm. Otherwise, it may converge to a local minimum.

2.2.2 Markov Random Field Approach

In [87], Li proposes feature based matching techniques based on Markov random fields to solve the problems for high level vision such as object matching and recognition and pose estimation. The features are extracted from images including critical points, lines, and surface patches. Li formulates the matching problem with a Maximum A Posteriori of a Markov Random Field, MAP-MRF, framework [84, 85]. For estimating the posterior probability in the framework of MAP-MRF, contextual constraints are captured in both prior and observation in an MRF formulation. The prior energy is defined in terms of both the single site potential and the pair site potential. The likelihood function is the conditional probability density function defined according to the observed data [86].

The MAP-MRF framework is also formulated to recognize multiple partially occluded objects by incorporating two kinds of contextual constraints: between object constraints and within object constraints. The matching between the scene and all the objects is determined by finding the maximum a posteriori solution over the admissible configuration space based on the observation and object models. Finding the solution is equivalent to minimising the posterior energy consisting of the prior and likelihood terms. The Hummel-Zucker relaxation labelling algorithm is used to solve the minimisation problem [63].

In [132], Starck and Hilton present a dense correspondence estimation approach for 3D sur-

faces undergoing free-form deformation. The surfaces are represented as triangulated meshes. The 3D surface correspondence is formulated as a problem to match surface points between two objects using local descriptors that are invariant to surface deformation. A locally isometric transformation that preserves geodesic surface distance is assumed for deformation. The deformations are inextensible, undergoing neither dilation or contraction during motion.

Feature descriptors are defined in terms of corner, edge, and region [132]. The 3D shape matching is performed as a probabilistic inference problem to seek the most likely labelling of correspondence with Markov random field that can help preserve the geodesic distance between feature points. Therefore, the correspondence labelling problem is transformed as a maximum a posterior (MAP) problem with Markov random field technique. The singleton potential function defines the likelihood of a correspondence label according to the distance between the corresponding feature descriptors. The compatibility function defines the prior probability for a realisation using the difference in the geodesic surface distance between the source points and the labelled assignment. Bayesian belief propagation is used to recover a maximum a posteriori (MAP) estimate for the joint distribution of correspondence labels. The final matching result is a dense surface to surface correspondence.

However, in Starck and Hilton's method, image property based feature descriptors defined in terms of corner, edge, and region are used for similarity measure. Firstly, these image property based feature descriptors are different from surface based shape descriptors such as curvature. Secondly, these feature descriptors defined in terms of corner, edge, and region are local descriptors, and are suitable for locally distinguishable objects. They are not suitable for neuroanatomical objects which do not have such locally distinguishable features.

In [6], Anguelov et al. propose a point-to-point dense correspondence recovery approach based on a Markov network and loopy belief propagation. The Markov network incorporates local surface geometry with spin images and geodesic distances between mesh points. Spin images are used as the local geometry similarity measure. Spin images offer an efficient way of comparing the surfaces around two points without requiring their rotational alignment, which is usually unknown. The spin images are compressed using principal component analysis

(PCA) to produce a low-dimensional signature of the local surface geometry around a point. Two low-dimensional signatures are compared simply by using their L_2 distance. The surface similarity potential is defined as a Gaussian distribution over the distance. The probabilistic inference over the Markov network is performed using loopy belief propagation.

In Anguelov et al.'s paper, the point-to-point dense correspondence is treated as non-rigid registration. This Markov network based point-to-point dense correspondence recovery approach is the closest literature to our method. However, our method is different from the approach in the following ways. Firstly, our method handles the matching and registration for a neuroanatomical object, the hippocampus. Usually, for neuroanatomical objects, their shapes are locally undistinguished. Anguelov et al. use spin images to be the local geometry similarity measure, while we explore Gaussian curvature. Secondly, our method can handle large surface meshes. To handle large surface meshes, computation complexity is a problem. We propose a computationally efficient algorithm to handle the problem, whereas in Anguelov et al.'s paper run-time is not mentioned. Thirdly, we explicitly present an algorithm for sparse update, whereas in Anguelov et al.'s paper a method for a nearness preservation constraint is mentioned, but no algorithm or details are given. Fourthly, we perform clamping in our method whereas Anguelov et al. do not. Clamping is a technique to increase the shape matching rate by integrating the message updating rule of belief propagation with the feedback of 3D rigid body registration.

2.3 Registration

Given two 3D objects, registration is to align the two objects by estimating a best transformation between them. Registration is divided into 3D rigid body registration, and non-rigid registration. 3D rigid body registration is to align 3D objects by estimating rigid body transformations including translation and rotation. One popular algorithm to achieve 3D rigid body registration is Iterative Closest Point (ICP). Non-rigid registration investigates techniques that deformably align one 3D object to another object. Non-rigid registration handles the alignment of data sets that are mismatched in a nonlinear or nonuniform manner, and it is a process to re-

cover the deformation and nonlinear transformation between two or more data sets. Non-rigid registration is also known as elastic registration. In this thesis, we investigate the recovery of dense correspondence between two 3D objects. The dense correspondence is obtained by 3D shape matching between two 3D objects formulated with a global framework Markov random fields. Dense correspondence can be used as a basis for rigid and non-rigid registration. As such we review the literature on rigid and non-rigid registration.

2.3.1 3D Rigid Body Registration

2.3.1.1 Spherical Harmonics Based Approach

Based on the first order spherical harmonics, Brechbuhler et al. [24] proposes a rigid body alignment method, called the first order ellipsoid (FOE). FOE alignment uses the fact that a shape reconstructed from the first order spherical harmonics forms an ellipsoid in object space. The FOE approach aligns the axes of ellipsoids to obtain the rigid body registration.

Alignment using FOE is effective in practice. FOE has been used to align 3D neuroanatomical objects while conducting the statistical and shape analysis of brain structures [47, 46]. However, it assumes that: (i) the shapes are pre-aligned to avoid the rotational and symmetry ambiguities of ellipsoids; and, (ii) they effectively can be approximated by an ellipsoid. For example, FOE alignment would not work with spherical or cuboid objects.

2.3.1.2 Procrustes Method

Procrustes method is based on a least squares type optimal fitting problem [58, 35, 50, 126]. Given two configurations of n non-coplanar points $A = \{a_i\}$ and $B = \{b_i\}$, Procrustes method seeks a transformation T which minimises

$$E(T) = \|A - T(B)\|.$$

Let d be the dimension of the coordinates of points a_i and b_i. A, B are n-by-d matrices whose rows are coordinates of points a_i, b_i, and $T(B)$ is the corresponding matrix of transformed

points. $\|\cdots\|$ is a matrix norm, for example the Frobenius norm $\|\cdots\|_F$. T could be affine or rigid body transformations.

Let T be a rigid body transformation, and replace the points of A and B with their corresponding points relative to their centroids, $a_i \mapsto a_i - \bar{a}$, $b_i \mapsto b_i - \bar{b}$, where $\bar{a}$ and $\bar{b}$ are centroids of A and B respectively. This reduces the Procrustes problem to an orthogonal Procrustes problem. For the orthogonal Procrustes problem, there is a condition to the above Procrustes method, $T^T T = I_d$, which means that T is orthogonal. Given A and B as defined above, the following algorithm can find an orthogonal d-by-d matrix T such that $\|A - BT\|_F$ is minimum [50]. Let $U^T XV = \Sigma = diag(\sigma_1, ..., \sigma_d)$ be the Singular Value Decomposition (SVD) of the matrix $X = B^T A$.

Algorithm 1 Orthogonal Procrustes

(1) $X = B^T A$;
(2) Compute the SVD $U^T XV = \Sigma$, and save U and V;
(3) $T = UV^T$.

Luo and Hancock use Procrustes method to perform point-set registration [94]. In [94] a method by iterating between Procrustes alignment and graph editing is proposed to improve the performance of the data alignment process. The Procrustes alignment method is applied to align the point-sets according to the estimate of the transformation parameters. The graph editing procedure is used to remove those nodes in the graph that are highly inconsistent. The process is iterated such that the accuracy of the date alignment can be improved. Wang and Mahadevan introduce an approach to use Procrustes method to perform manifold alignment of data sets [144]. The approach involves a mapping of the data sets from high dimensional spaces to low dimensional spaces reflecting the data sets' intrinsic geometries by using standard dimensionality reduction methods. After dimensionality reduction, Procrustes method is used to align the two low dimensional embeddings of the data sets based on a number of landmark points. By using Procrustes method, the translational, rotational and scaling components are removed from the two sets, and therefore the optimal alignment between the two sets can be obtained. During alignment, the manifold shape remains unchanged. However, Procrustes method has two shortcomings [95]. Firstly, it requires that the point sets to be aligned have

identical size. Secondly, it requires that the points should have been labelled, i.e. the correspondences to be known in advance. The two shortcomings limit its application in point set alignment.

2.3.1.3 Iterative Closest Point (ICP)

For 3D rigid body registration, another popular approach is the Iterative Closest Point (ICP) algorithm. The algorithm has been first introduced by Besl and McKay [19]. The ICP algorithm is a representation-independent method for the registration of 3D shapes including free-form curves and surfaces. Given two independently acquired sets of 3D points, M (model set) and D (data set) which correspond to the same shape, the aim of ICP is to find a transformation consisting of a rotation R and a translation t which minimises the following cost function,

$$E(R,t) = \sum_{i=1}^{|M|} \sum_{j=1}^{|D|} w_{i,j} \left\| m_i - (Rd_j + t) \right\|^2,$$

where m_i and d_i are the points in model set and data set respectively; and $w_{i,j}$ are the weights for point matches. The weights are assigned as follows: $w_{i,j} = 1$, if m_i is the closest point to d_j; $w_{i,j} = 0$, otherwise.

The ICP algorithm registers the data set D to be in best alignment with the model set M. The main steps include: (1) compute the closest points; (2) compute the registration; (3) apply the registration. The ICP algorithm iteratively calculates the point correspondences. At each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation (R,t) for minimising the above cost function. The assumption is that in the previous iteration step the point correspondences are correct. Besl and McKay [19] prove that the ICP algorithm always converges monotonically to a local minimum with respect to the above mean-square distance cost function. Although the ICP algorithm must converge monotonically to a local minimum from any given rotation and translation of the data point set, it may or may not converge on the desired global minimum [19].

Chen and Medioni [30] enhanced the basic algorithm by assuming that the model points m_i can be supplied with surface normals $\mathbf{n_i}$, which allows the point-to-point distance to be

replaced by a point-to-tangent plane. The enhancement gives improved convergence properties on the basic algorithm without a significant increase in computational cost [42].

ICP is efficient, with average case complexity of $O(n \log n)$ for n points. However, a good initial estimation of the transformation is required for convergence to the global optima.

2.3.1.4 Rigid Body Registration Based on Deformation

Li et al. [83] present a rigid body registration approach based on an active surface model, which can be used for surface-to-3D images and surface-to-surface registration. In each iteration, the surface is deformed first through partial differential equations, and then the least-squares estimate of the rigid body transformation between the deformed surface and predefined surface is computed. The surface deformation is performed by minimising an energy functional containing the internal and external energy terms. The internal energy constrains the smoothness of the model, whereas the external energy attracts the active surface model to the features of interest. Singular value decomposition is used to estimate the rigid body transformation. The surface deformation guides the rigid body transformation. The limitation of the method is that the deformation step needs a proper way to define the external energy term from the properties of 3D images such as image intensity to guide the deformation process [67].

FOE alignment does not handle general objects such as spherical or cubic objects, and it also needs pre-alignment. Procrustes method, Iterative Closest Point algorithm, rigid body registration based on deformation, all the other rigid body registration methods surveyed here suffer from the convergence to local minimum during the optimisation. Contrary to this, our MRF based 3D rigid body registration can reach the global minimum.

2.3.2 Non-Rigid Registration

The purpose of non-rigid registration is to deformably align objects. Currently a key approach to non-rigid registration is based on diffeomorphic mappings. We will review diffeomorphism based non-rigid methods including LDDMM, Dartel, and phase mutual information. The previous non-registration methods include elastic deformation [13], viscous fluid deformation

[80], optical flow [62], finite element method (FEM) [153], thin-plate splines [21], and B-splines [78]. We will not review these methods here. We will focus on diffeomorphism based methods, and a global Markov net based approach and non-rigid ICP.

2.3.2.1 Diffeomorphic Non-rigid Registration

Diffeomorphism is invertible, continuous and one-to-one mapping for a given image. What invertibility, continuity and one-to-one mappings mean is that for each transformation, (1) the transformation has an inverse so that any transformation (or warp) can be reversed; (2) every point must move to give a continuous flow of intensities. The transformation should not corrupt structures in the image; (3) no two points should be mapped to the same point as this would "strip off" areas in the image. The beautiful mathematical framework of diffeomorphism supports its popularity among the non-rigid registration methods.

LDDMM The Large Deformation Diffeomorphic Metric Mapping (LDDMM) is a framework for diffeomorphic registration in computational anatomy [15]. In the LDDMM framework, transformations are assumed to belong to a group of diffeomorphisms endowed with a Hilbert differentiable structure of Riemannian manifold [140].

Based on the LDDMM, Qiu et al. [120] propose a parallel transport in diffeomorphisms approach to distinguish subjects with very mild Alzheimer's disease (AD) and subjects with preclinical forms of AD from healthy comparison subjects using time-dependent patterns of hippocampal surface deformation. The parallel transport is based on the LDDMM, which is used to represent within-subject deformation in the global template coordinates [120, 119]. The parallel transport operation is taken from Riemannian geometry, which displaces vectors along a curve without changing properties such as norms of the vectors or their dot products. In Euclidean space, this operation is the standard translation of vectors. In curved spaces, parallel translation is nonlinear and can be calculated by solving a differential equation [120]. The parallel transport in diffeomorphisms allows the deformation analysis to have a common coordinate system across subjects to perform hypothesis tests related to time-dependent within subject deformation.

In [120], Qiu et al. perform three-level analyses under the LDDMM framework to compare longitudinal shape variation of the hippocampus across clinical populations. The first two levels assess deformations within subjects and between a global template and subjects respectively by using the LDDMM. The surface of the hippocampus is represented by a triangulated mesh. The use of LDDMM for studying the shapes of objects provides a diffeomorphic transformation, and defines a metric distance that can be used to quantify the similarity of two shapes. In [120], the LDDMM surface mapping algorithm is used to transform one hippocampal surface to another and incorporate intrinsic geometry of the surface such as the normal vector which can be used to quantify the similarity of surface shapes. The third level analysis uses parallel transport in diffeomorphisms to translate within subject deformation between time points in a global template without incorporating across subject deformation. The optimisation of LDDMM is performed by using Gauss-Newton method.

DARTEL In [9], Ashburner proposes a diffeomorphic image registration method, called the diffeomorphic anatomical registration using exponentiated Lie algebra (DARTEL). The DARTEL model assumes a flow field that remains constant over time. The flow field refers to a velocity field of a moving fluid. In this case, a velocity vector is associated to each point in the fluid. A deformation in terms of the flow field can be described as a differential equation. The Euler method is used for integration. The Euler method is for numerical integration of the first order differential equation. The flow field can be considered as an element of the Lie algebra, which is exponentiated to generate deformations [9]. Since the Jacobian of a deformation that conforms to an exponentiated flow field is always positive, this ensures the deformation is diffeomorphic and invertible. Therefore, both forward and inverse transformations can be produced from the same flow field. A typical flow field is exponentiated to produce a forward deformation, and the negative of the flow field is exponentiated to produce the inverse deformation.

The non-rigid registration involves simultaneously minimising a measure of difference between the image and the warped template, while also minimising an energy measure of the deformations used to warp the template. The objective function for the optimisation problem

consists of a prior term and a likelihood term. The prior term incorporates the prior probability of the parameters with respect to the deformation. The discrete parameterisation of the flow field is by a linear combination of basis functions such as B-splines. For the implementation of DARTEL, the prior term can take a variety of forms including membrane, bending, and linear elastic energy. The likelihood term incorporates the probability of the image data given the parameters.

Optimisation is performed by using a Levenberg-Marquardt (LM) approach. The LM algorithm uses an iterative scheme to update the parameter estimates. At each iteration, the first and second derivatives of the objective function with respect to the parameterisation of the deformation are calculated. A full multigrid (FMG) approach [53] is used to solve the update equations during the optimisation. The FMG approach is based on relaxation methods, which are performed at multiple scales in order to enhance the speed. Compared to variable velocity diffeomorphic models such as LDDMM, the DARTEL has advantages in terms of the speed of execution. The variable velocity means that the velocity in the velocity field associated with diffeomorphic models is not constant over time.

In [9], the DARTEL registration approach is evaluated in terms of how well the resulting deformations encode the shape information necessary to separate male and female subjects and to predict the ages of the subjects. The idea is that DARTEL should encode relevant and important anatomical features. Classification performance can be used as a surrogate measure of the quality of the features encoded by DARTEL. To demonstrate this validation approach, support-vector machines were used to classify images according to sex, and relevance-vector machines to estimate the ages of subjects based upon their images. This involved training with a random selection of 400 of the subjects, and then making predictions about the remaining 71 subjects.

Phase Mutual Information In [160] Zhang presents a new non-rigid registration method to register a real-time 3D ultrasound volume slice to a cardiovascular magnetic resonance image. The non-rigid registration is based on phase mutual information [101]. Phase mutual information is obtained by replacing intensity for intensity mutual information with local phase.

Phase is the angle of a signal in terms of standard complex representation while performing the Fourier transform for an image. It is a local feature descriptor in the frequency domain. The corresponding joint entropy, marginal entropy, and mutual information are obtained based on local phase respectively. Phase mutual information makes the assumption of the independence of the feature types between the modality pairing, so that two features with considerably different appearances can be successfully aligned and compared with each other.

In [160] Zhang demonstrates that phase mutual information is better suited to register a pair of multimodal image than intensity because by using local phase as a image descriptor image structure rather than signal magnitude can be better described. It has been shown in visual perception experiments that local phase contains more perceptually significant major part of the information in a signal. An image can be decomposed into amplitude and phase information. The local phase is shown to be uniquely defined. The local phase represents most of the useful image information, whereas the local amplitude provides a measure of the confidence of the extracted phase. The local phase can be interpreted as a qualitative description of a detected feature in a signal, such as the edge or ridge in a signal [101]. Phase mutual information can therefore detect relationships between local shapes, enabling it to capture the relationship between two images.

The local deformation of the non-rigid registration is modelled by a polyaffine transformation [7]. The main idea of the polyaffine transformation is to combine multiple affine transformations to form a diffeomorphic mapping. The polyaffine transformation is a parametric transformation that exhibits a locally affine behaviour and has invertibility property, which ensure the mapping is a diffeomorphism. The control points used in the polyaffine transformation are detected automatically and refined by calculating a local mis-match measure based on phase mutual information. The polyaffine transformation is modelled by an ordinary differential equation (ODE). Since all transformations induced by an ODE are reversible, the polyaffine transformation obtained via an ODE can guarantee the invertibility of the transformation.

Application of Diffeomorphic Non-rigid Registration – Atlas Creation A human brain atlas is an atlas of neuroanatomy structures of the human brain. Abants and Gee propose an

atlas creation algorithm by using non-rigid image registration techniques in [10]. According to Abants and Gee [10, 11, 12] computerised atlases based on MRI images may capture either average shape, average intensity or both in a single image. In [10], Avants and Gee illustrate that the average anatomical shape atlas should be computed with respect to the image registration problem. An extension of the group theoretical diffeomorphic registration algorithm from [104] is used as the framework for the atlas creation in [10]. In computational anatomy, the group of diffeomorphisms gives one-to-one mappings between the objects in a collection. The diffeomorphisms are C^n (n times continuously differentiable) smooth, invertible, topology preserving transformations.

The individual mappings can be found by using a variety of non-rigid image registration algorithms. The transformation model for the non-rigid registration is established by a one-parameter diffeomorphism group. The transformation of anatomical images with respect to the diffeomorphism group is obtained through the time dependent action generated by velocity fields or flows [10]. The mapping is generated from an ordinary differential equation. The group formulation can be used to measure distances between anatomical instances with a true mathematical metric. Via the group action, the shortest path between instances of the anatomy can be found. The non-rigid registration and metric are obtained through solving a variational problem. The Euler-Lagrange equations for the variational problem were derived in [103]. The variational problem is solved by using the finite element method. Abants and Gee illustrate and compare traditional linear averaging and geodesic averaging approaches for atlas estimation in [10].

Summary of Diffeomorphic Non-rigid Registration In summary, for LDDMM the deformation is realized by the flow of a time-dependent velocity vector field, whereas for DARTEL a flow field that remains constant over time is assumed. By assuming a constant flow field, DARTEL can speed up the execution time. Despite the solid foundation of the mathematical framework of LDDMM, the high computational requirements have made this methodology less attractive for clinical applications where more efficient registration algorithms are usually preferred [57].

2.3.2.2 Markov Net

In Section 2.2.2, we presented Anguelov et al.'s paper as a method for finding point-to-point dense correspondence. It also claims to be a method for non-rigid registration. The non-rigid registration of two meshes is realized by optimising a joint probabilistic model over all point-to-point correspondence between them. The model enforces preservation of local mesh geometry by spin images, as well as more global constraints that capture the preservation of geodesic distance between corresponding point pairs. The task of registration is to estimate the set of all correspondences and a non-rigid transformation which aligns the corresponding points. Each point in the mesh is attached to a local coordinate system. For each point, a non-rigid transformation defines a translation of its coordinates and a rotation of its local coordinate system. The non-rigid method has been successfully used to register two surfaces of an object undergoing articulated movement and non-rigid deformation [6].

2.3.2.3 Non-rigid ICP

In [6], point-to-point dense correspondence is treated as a basis for non-rigid registration. The final non-rigid registration is realized by combining point-to-point dense correspondence with non-rigid ICP [54]. In [54], Hahnel et al. propose a non-rigid ICP algorithm by extending the standard ICP algorithm for modelling non-rigid objects with mobile robots. The non-rigid ICP algorithm is obtained by replacing rigid links between the measurement coordinates and the robot sensor in standard ICP with soft links between adjacent measurement points. The resulting problem of registration under these soft constraints becomes a high-dimensional optimisation problem with more variables involved than in standard ICP. The problem is solved by using Taylor series expansion and a coarse-to-fine hierarchical technique.

Similarly, our MRF based point-to-point dense correspondence recovery method can be used as a basis for non-rigid registration. By combining with non-rigid ICP, the explicit non-rigid registration can be realized. We do not perform non-rigid ICP by combining with our dense correspondence recovery method in this thesis. This will be our future work.

2.3.2.4 Summary of Non-rigid Registration

In LDDMM and DARTEL, optimisation is performed by using Gauss-Newton and Levenberg-Marquardt (LM) methods respectively. For Gauss-Newton and Levenberg-Marquardt methods, they are prone to local minima and a good initialisation is needed for the final solution. For similar reasons, most of the non-rigid registration methods need good initialisation and may converge to a local minimum. The Markov net based approach proposed by Anguelov et al. is an exception in that it does not require spatial initialisation. We have discussed the difference of our method with Anguelov et al.'s in Section 2.2.2.

2.4 Global Methods for Energy Function Minimisation

As described in Chapter 10 of [87], the global methods for energy function minimisation mainly include Graduated Nonconvexity (GNC), Belief Propagation, Graph Cuts, Genetic Algorithms, Mean Field Annealing, and Simulated Annealing.

Graduated Nonconvexity (GNC) is a deterministic approach to obtain an approximate global solution for nonconvex continuous minimisation problem [20]. The basic idea is to set the control parameter to a sufficiently large value such that the energy function is strictly convex. The minimum is found by using the gradient-descent method, which is used as the initial value to find the minimum in next phase with a lower value for the control parameter. If the control parameter is decreased, the energy function becomes nonconvex and local minima appear. By tracking the sequence of minima, the approximate global solution would be obtained.

Graph Cuts is another global method for minimising the energy function arising from MRF modelling. It was initially proposed to solve the minimisation problem for the Ising MRF model, a two-label MRF problem [51]. The global solution may be obtained by using max-flow/min-cut algorithms. For a multiple-label problem, if the smoothness term of the energy function is convex, then the multiple-label problem may be transformed to a two-label problem. The exact solution would be obtained. For a multiple-label problem with more general energy function, the minimisation is NP-hard. Boykov et al proposed two algorithms, α-expansion

and α-β swap, to approximate the global solution [22]. The graph cuts algorithm basically does one label at a time and tries to find the boundary between labels.

Mean Field Annealing is a special relaxation labelling algorithm combined with an annealing process to minimise the energy function [117]. The temperature is used to control the annealing process for the energy function defined based on the Gibbs distribution. As the temperature decreases to zero, the mean of the Gibbs distribution approaches its mode or the global minimum of the energy function. The key part of the Mean Field Annealing is the calculation of the partition function as once the partition function is obtained all the statistical information about the system can be deduced from it. Genetic Algorithms (GAs) are a method very similar to annealing, which were derived from the evolutionary process in biology [59, 48]. The basic principle is that the best fit survives. The minimisation process includes chromosome generation, evolution, crossover, mutation, and fitness calculation.

Belief Propagation is the main focus of this thesis, and we will discuss Simulated Annealing in Chapter 6.

2.5 Summary

In this chapter, we have performed a literature review on 3D shape matching, dense correspondence, rigid and non-rigid registration, and global methods for energy function minimisation. From the literature review, we see that there is demand for literature on globally handling shape matching issue and generating local point-to-point dense correspondence. The global framework proposed in this thesis based on Markov random fields by combining with curvature for shape matching can fulfil the literature gap.

Chapter 3

Background

In this chapter, we review the background material upon which the contributions of this thesis are based. Since we solve the 3D shape matching with Markov random fields and belief propagation, we provide background mathematics on Markov random fields and belief propagation. We use curvature at a particular scale as a similarity measure while performing the 3D shape matching. Therefore, we provide background on curvature and scale space as well. In addition, we present 3D rigid body transformation estimation techniques based on three pairs of corresponding points. Our surface based rigid body registration is used to register human hippocampi. For completeness, we supply background on the hippocampus and MRI.

3.1 Markov Random Fields

In this section, we present an overview on Markov random fields from [25]. Please see [25] for an introduction on Markov random fields. Suppose M is a finite index set $\{1, 2, ..., n\}$. Let $\{X_i, i \in M\}$ be a finite collection of random variables with X_i taking values in a finite set of states S.

The vector $X = (X_1, ..., X_n)$ takes values in $\Omega = S^n$, the set of all n-tuples $x = (x_1, ..., x_n)$ with $x_i \in S$ for each i. We call Ω the *configuration* space. From subset $Y \subseteq X$ we define a configuration y of Y with respect to a state space S as one realisation x_i, $x_i \in S$ and $X_i \in Y$.

For a given random variable X_i being in a state x_i, its conditional probability is written as

$$P(X_i = x_i | \{X_k\}, k \neq i).$$

It means that the probability of X_i being in a state x_i depends on the values of all of the other random variables $\{X_k\}$. Instead, if this probability only depends on some, but not all the random variables, then we say that the collection of the random variables has the *Markov property*. Let $N(i)$ denote the subset of $\{X_k\}$ on which X_i depends, we can write this dependence as

$$P(X_i = x_i|\{X_k\}, k \neq i) = P(X_i = x_i|\{X_k\}, k \in N(i)).$$

Any collection of random variables having the Markov property can be referred to as a Markov network.

Let us call the set $N(i)$ the neighbourhood of X_i. Let us define a neighbourhood system on M be a family $N_M = \{N(i)\}_{i \in M}$ of subsets of M such that for all $i \in M$ [25],

(i) $i \notin N(i)$,

(ii) $j \in N(i) \Rightarrow i \in N(j)$.

We may represent the neighbourhood system on M by a graph whose nodes are the nodes of M and whose edges are defined by N_M: nodes i and j are linked by an edge if and only if they are neighbours, i.e., $j \in N(i)$.

Let $\tilde{N}(i)$ denote the set of node i and its neighbours, $N(i) \cup \{i\}$. Let $X(u)$ denote the set of random variables with indices in set u. The random field X is called a *Markov random field (MRF)* with respect to the neighbourhood system N_M if for all nodes $i \in M$, the random variable X_i and $X(M \backslash \tilde{N}(i))$ are independent given $X(N(i))$.

3.1.1 Cliques, Potential, and Gibbs Distributions

Let us define a graph specified by the finite set M and its neighbourhood system N_M as (M, N_M) where vertices are defined by M and edges are defined by N_M. We define a clique C of the graph (M, N_M) as any subset of nodes $C \subseteq M$ either with a single node $\{i\}$ which is called a singleton, or with more than one element such that for any two distinct nodes i and j of C, i and j are neighbours. A clique C is called maximal if for any node i, $C \cup \{i\}$ is not a clique.

A potential is a function indexed by subsets of M on the configuration space Ω. We write potential as $V_C(x)$ for $C \subseteq M, x \in \Omega$. The potential $V_C(x)$ returns a real value for each clique C over the configuration x.

Given a set of all potentials, the energy of a configuration x is the sum of clique potentials V_C over all cliques

$$E(x) = \sum_C V_C(x).$$

The distribution

$$P(x) = \frac{1}{Z} e^{-\frac{1}{T}E(x)} \tag{3.1}$$

is called a *Gibbs distribution* on the configuration space Ω [72], where $T > 0$ is the temperature, $E(x)$ is the energy of configuration x, and Z is the normalising constant

$$Z = \sum_{x \in \Omega} e^{-\frac{1}{T}E(x)},$$

called the *partition function*.

$P(x)$ takes it values in $[0, 1]$, and

$$\sum_{x \in \Omega} P(x) = 1.$$

A set of random variables X on M is said to be a Gibbs random field (GRF) on M with respect to neighbourhood system N_M if and only if its configurations obey a Gibbs distribution as defined above.

An MRF is characterised by its local property (the Markov property) whereas a GRF is characterised by the Gibbs distribution. The Hammersley-Clifford theorem establishes the equivalence of the Markov property and the Gibbs distribution. The theorem states that X is an MRF on M with respect to N_M, if and only if X is a GRF on M with respect to N_M.

Theorem 3.1.1 (Hammersley, 1968). *Let P be the distribution of a Markov random field with*

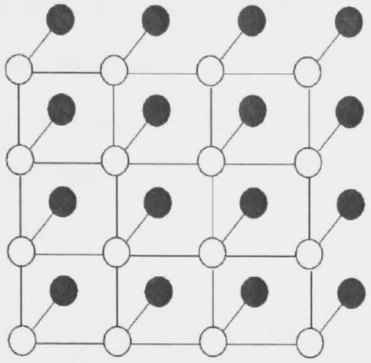


Figure 3.1: A square lattice pairwise Markov random field. Figure from Yedidia et al. [155].

respect to a configuration satisfying the positivity condition. Then [152]

$$P(x) = \frac{1}{Z} e^{-H(x)} \tag{3.2}$$

for some energy function $H(x)$ deriving from a Gibbs potential $\{V_C\}C \subset X$ associated with the topology of x.

3.2 Belief Propagation

Belief Propagation (BP) is a popular algorithm for approximate inference on graphical models including Markov random fields.

One difficult task in computer vision is to find theoretically solid models that are computationally tractable. Pairwise Markov random fields (MRFs) provide attractive theoretical models for some computer vision problems [155]. These computer vision problems include stereo matching, optical flow, and image restoration.

In [155], Yedidia et al. describe a square lattice pairwise Markov random field for images. A graphical depiction of this model is presented in Figure 3.1.

Let i be a single pixel in an image. From [155], given pixel intensities about the image y_i, the task is to infer some other quantities about the underlying scene x_i. Considering the statistical dependency between x_i and y_i at each position i, let us define a joint compatibility

function between x_i and y_i as $\phi_i(x_i, y_i)$, which is usually called the *"evidence"* for x_i. At the same time, since the nodes i are arranged in a two-dimensional grid, let us define the compatibility function between the scene variable x_i and its neighbour scene variable x_j as $\psi_{ij}(x_i, x_j)$; ψ_{ij} only connects neighbour positions. Therefore, the overall joint probability of a scene x_i and an image y_i is [155]

$$P(x) = \frac{1}{Z} \prod_{(ij)} \psi_{ij}(x_i, x_j) \prod_i \phi_i(x_i, y_i)$$

where Z is a normalisation constant and the product over (ij) is over nearest neighbours on the square lattice.

The inference problem related to this model is to calculate the belief $b(x_i)$ at each node i, and then based on the beliefs to infer some quantities about the underlying unknown scene.

Let us write $\phi_i(x_i)$ as a short-hand for $\phi_i(x_i, y_i)$. The joint probability distribution for the unknown variable x_i is then

$$P(x) = \frac{1}{Z} \prod_{(ij)} \psi_{ij}(x_i, x_j) \prod_i \phi_i(x_i). \tag{3.3}$$

Let $m_{ij}(x_j)$ be the message from node i to node j about what state node j should be in, then the corresponding message updating rule is [154]

$$m_{ij}(x_j) \leftarrow z \sum_{x_i} \phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \backslash j} m_{ki}(x_i). \tag{3.4}$$

where z denotes a normalisation constant, and $N(i)\backslash j$ is the set of all nodes neighbouring node i except j. Note that on the right-hand-side, we take the product over all messages going into node i except for the one coming from node j.

The belief at a node i is proportional to the product of the local evidence at that node $\phi_i(x_i)$, and all the messages coming into node i [155]

$$b_i(x_i) = z \phi_i(x_i) \prod_{k \in N(i)} m_{ki}(x_i) \tag{3.5}$$

where z is the same as defined above and $N(i)$ denotes the nodes neighbouring i. The beliefs must sum to 1.

As shown in [114], for a graph without cycles, the inference by belief propagation is exact. It is guaranteed that the calculated beliefs will converge to the exact posterior distribution. Although there is no theoretical proof to guarantee the convergence while applying belief propagation to a graph with loops, it has been widely reported that it works well in many practical problems, such as stereo matching [134, 137] and image restoration [38] for early vision.

In addition, as shown by Kschischang et al. (2001) with factor graph theory in [77], a certain number of algorithms in artificial intelligence, signal processing, and digital communications can also be derived from the belief propagation algorithms, including the well known Viterbi algorithm and Kalman filter. It has also been reported that similar ideas to BP have also been successfully used for codes, especially turbo codes. McEliece et al. (1998) [100] have shown the equivalence between belief propagation and the turbo decoding algorithm introduced by Berrou et al. [18] in 1993. As stated by McEliece et al. in [100], the decoding algorithm is regarded as one of the most exciting and potentially important developments in coding theory in many years.

Previous work on using belief propagation for shape matching includes non-rigid registration of two surfaces in 3D [6] and matching deformable template models to objects in an image in 2D [32].

3.3 Intrinsic Geometric Properties of the Gaussian Curvature

Curvature is a local measure of geometric properties, and can be used to represent local shape information. We focus on Gaussian curvature, as it is one of the fundamental second order geometric properties of a surface. According to Gauss's Theorema Egregium [131], Gaussian curvature is intrinsic. For a local isometric map $f : A \rightarrow A'$ between two surfaces, it remains invariant. Therefore, in this thesis we will use Gaussian curvature as a criterion for similarity measure while performing the 3D shape matching between two 3D objects by using Markov random field techniques, which will be introduced in Chapter 4. The representation satisfies

invariance, uniqueness, and stability criteria required for a shape descriptor as described in [108].

3.3.1 Gaussian Curvature Estimation

Let T_pA be the tangent space of a manifold A at point p, and P_{II} be a two-dimensional subspace of T_pA. The curvature κ of A at a point $p \in A$ is interpreted as a map [151]

$$\kappa : P_{\text{II}} \to \mathbb{R}.$$

The first and second fundamental forms of A are defined as follows [131]

$$I(p)(U_p, V_p) = \langle U_p, V_p \rangle,$$
$$II(p)(U_p, V_p) = -\langle \nabla_{U_p}\mathbf{n}, V_p \rangle,$$

where $U_p, V_p \in T_pA$, $\mathbf{n}$ is the normal vector at the point p, $-\nabla_{U_p}\mathbf{n} : T_pA \to T_pA$ is the Weingarten map, and $\langle \cdot, \cdot \rangle$ is the Riemannian metric on the manifold A.

Suppose a local parameterisation for $A \subset \mathbb{R}^3$ is a coordinate patch $\sigma(\mu, \nu) : \Omega \to A$, where $\Omega \subset \mathbb{R}^2$ is an open domain, then a surface can be defined as a map from $\mathbb{R}^2$ to $\mathbb{R}^3$, $\sigma(\omega) = (x(\mu, \nu), y(\mu, \nu), z(\mu, \nu))^T \subset \mathbb{R}^3$, where $\omega \in \Omega$. Given vectors σ_μ and σ_ν with respect to μ and ν respectively, the normal vector on the surface may be obtained locally by [55]

$$\mathbf{n}(\mu, \nu) = \frac{\sigma_\mu \times \sigma_\nu}{\|\sigma_\mu \times \sigma_\nu\|}.$$

Let the surface metric be $ds^2 = \sum_{\mu,\nu} g_{\mu\nu} d\mu d\nu$ where $g_{\mu\nu}$ is a symmetric covariant tensor of the second order, and $\mathbf{n}$ be the unit surface normal as defined, then the corresponding first and second fundamental forms for σ are $Ed\mu^2 + 2Fd\mu d\nu + Gd\nu^2$ and $ed\mu^2 + 2fd\mu d\nu + gd\nu^2$, where, E, F, G, are $\sigma_\mu \cdot \sigma_\mu$, $\sigma_\mu \cdot \sigma_\nu$, and $\sigma_\nu \cdot \sigma_\nu$; and e, f, g are $\sigma_{\mu\mu} \cdot \mathbf{n}$, $\sigma_{\mu\nu} \cdot \mathbf{n}$, and $\sigma_{\nu\nu} \cdot \mathbf{n}$ respectively. The principal curvatures k_1 and k_2 at a point p are the eigenvalues of the following

Weingarten curvature matrix [131]

$$W = \begin{bmatrix} e & f \\ f & g \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1}.$$

The Gaussian curvature $\kappa = k_1 k_2$ is the product of the principal curvatures at the point p. It is estimated based on the principal curvatures using local charts.

For a surface mesh, a local 3D coordinate frame is formed with its origin at the vertex, and its coordinate axes as the normal vector and two arbitrary orthogonal axes in a plane perpendicular to this vector.

3.3.1.1 Curvature Estimation by Using Covariance Matrix

In this thesis, we use a curvature estimation technique by using covariance matrix proposed by Berkmann and Caelli to estimate the Gaussian curvature [17, 28].

The local first order covariance matrix at a point $\mathbf{x}$ of the surface is defined as

$$C_I = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x_i} - \mathbf{x_m}) \cdot (\mathbf{x_i} - \mathbf{x_m})^T,$$

where $\mathbf{x_i} = (x_i, y_i, z_i)$ are the points in a neighbourhood of $\mathbf{x}$, $\mathbf{x_m} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x_i}$ is the mean position vector, and n is the number of points in the chosen neighbourhood of $\mathbf{x}$.

The eigenvalues and eigenvectors of C_I determine three orthogonal vectors. The eigenvectors corresponding to the tangent plane are chosen as those which maximise the variance of the point projections onto the three possible planes.

The second-order covariance matrix can then be calculated from the projection of the normal vectors onto the tangent plane in $\mathbf{x}$. The eigenvectors of this two-dimensional matrix correspond to the principal directions as defined in classical differential geometry, whereas the eigenvalues give us a measure of the principal curvatures of the surface in those directions. The Gaussian curvature can be obtained from the estimated principal curvatures.

3.3.1.2 Local Surface Curvature Estimation

Based on mesh representation of a surface with spherical topology, Delingette et al. [34] proposes a robust measure of curvature computed at every node of a mesh from the relative positions of its three neighbours. The topology constraint on the mesh requires that each vertex has three neighbouring vertices in order to form a tetrahedron at each vertex. The mesh is computed from the data by deforming a standard shaped mesh, for example, an ellipsoid, until it fits the surface of an object. The method is robust mainly because all the nodes are at a relatively stable position after the deformation process. The deformable surface process serves as a smoothing operation over the possibly noisy original data [127]. This measure of curvature proposed by Delingette is called the simplex angle, which varies between $-\pi$ and π. Negative means concave, and positive means convex. This behaviour of the simplex angle is consistent with the intuitive notion of local curvature of a surface. For a configuration of four points, the simplex angle only depends on the relative positions of the points, not on their absolute positions. By using the simplex angle, the local curvature at each node can be calculated by its position relative to its neighbours.

3.3.2 Curvature in Scale Space

The multi-scale representation of a signal is to generate a one-parameter family of derived signals where fine-scale information is successively suppressed [91]. One methodology proposed to obtain such a multi-scale representation of a measured signal is by embedding the signal into a one-parameter family of derived signals, the scale-space, where the parameter, denoted scale parameter $\tau \in \mathbb{R}_+$, is intended to describe the current level of scale [91].

Given a N-dimensional continuous signal $h : \mathbb{R}^N \rightarrow \mathbb{R}$, the scale-space representation $Q : \mathbb{R}^N \times \mathbb{R}_+ \rightarrow \mathbb{R}$ is defined such that the representation at "zero scale" is equal to the original signal

$$Q(\cdot; 0) = h.$$

The representation at coarser scales are given by convolution of the given signal with Gaus-

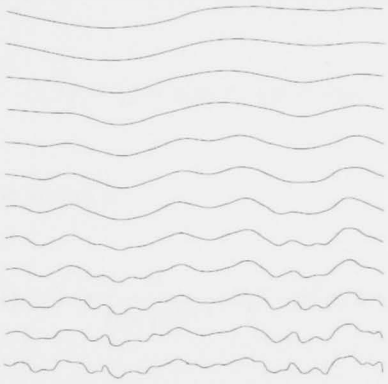


Figure 3.2: Applying Gaussian smoothing to a one-dimensional signal. Figure from Tony Lindeberg [91].

sian kernels of successively increasing width [91]

$$Q(\cdot;\tau) = g(\cdot;\tau) * h.$$

With the expression of explicit integrals, the result of the convolution operation $*$ can be expressed as

$$Q(x;\tau) = \int_{\xi\in\mathbb{R}^N} g(\xi;\tau)h(x-\xi)d\xi,$$

where $x = (x_1, \cdots, x_N)^T \in \mathbb{R}^N$, and $g : \mathbb{R}^N \times \mathbb{R}_+\backslash\{0\} \to \mathbb{R}$ is the N-dimensional Gaussian kernel. The result of applying Gaussian smoothing to a one-dimensional signal is shown in Figure 3.2.

Curvature is naturally associated with scale space, and it manifests different values in different scales. In order to estimate curvatures under different scales, the original surfaces are smoothed first with a smooth kernel with different variances. Curvatures are then calculated based on the smoothed surfaces accordingly. The different sizes of the mask will generate curvatures at different scales.

Let m be a surface. Its linear scale-space representation is a family of derived surfaces

$L(\tau)$ defined by convolution of p, $p \in m$, with a smooth kernel $g(p;\tau)$ [91],

$$L(\tau) = \{p * g(p;\tau) \mid p \in m\}.$$

In the case of Gaussian, the kernel is

$$g(p;\tau) = \frac{1}{(2\pi\tau)^{N/2}} exp(-p^T p/2\tau),$$

and τ is referred to as the scale parameter. The square root of the scale parameter τ is the standard deviation of the Gaussian kernel.

It has been shown that the solution of the scale-space family can be obtained equivalently by solving the diffusion Equation [73],

$$\frac{1}{2}\Delta L = \frac{1}{2}\sum_{i=1}^{N} \partial_{p_i p_i} L = \partial_t L,$$

with initial condition $L(0) = m$, the original surface.

The Gaussian curvature κ at a point p on a derived surface L is a function of the scale parameter τ, corresponding to different scales upon choosing different τ.

3.4 3D Rigid Body Transformation Estimation from Three Pairs of Corresponding Points

In this thesis, we consider two ways of obtaining the 3D rigid body transformation from three pairs of point correspondences. One is the coordinate system construction based three-points algorithm suggested by Horn [60], and the other is the critical point based algorithm proposed by Liu and Rodrigues [92]. The coordinate system construction based approach establishes two coordinate systems containing each three corresponding points respectively first, and then obtains the rotation matrix by aligning the two coordinate systems. The critical point based algorithm estimates rotation matrix by first constructing the rotation axis from the three pairs of correspondences, then determining the critical point by the projections of two pairs of corre-

sponding points and two non-parallel correspondence vectors, and finally obtaining the rotation angle from the critical point.

3.4.1 Coordinate System Construction Based Approach

We discuss establishing correspondences between two coordinate systems, which, by convention, we will call left and right respectively. Let the three points in each of the two coordinate systems be $v_{l,1}, v_{l,2}, v_{l,3}$ and $v_{r,1}, v_{r,2}, v_{r,3}$ respectively. Suppose three point correspondences are given with $v_{l,1}, v_{l,2}, v_{l,3}$ corresponding to $v_{r,1}, v_{r,2}, v_{r,3}$. According to Horn's closed form solution of translation and rotation [60], the unit vectors of the left coordinate system can be constructed from:

$$\begin{aligned}
\hat{x}_l &= \frac{v_{l,2} - v_{l,1}}{\|v_{l,2} - v_{l,1}\|_2}, \\
\hat{y}_l &= \frac{(v_{l,3} - v_{l,1}) - [(v_{l,3} - v_{l,1}) \cdot \hat{x}_l]\hat{x}_l}{\|(v_{l,3} - v_{l,1}) - [(v_{l,3} - v_{l,1}) \cdot \hat{x}_l]\hat{x}_l\|_2}, \\
\hat{z}_l &= \hat{x}_l \times \hat{y}_l.
\end{aligned}$$

The unit vectors of the right coordinate system can be established in the same way.

Let $R \in SO(3)$ be the rotation matrix, $t \in \mathbb{R}^3$ be the translation vector, and $s \in \mathbb{R}$ be the scaling factor.

After aligning the unit vectors of the left and right of the two coordinate systems, the rotation matrix R can be obtained by:

$$R = U_r U_l^T,$$

where U_l and U_r are orthonormal matrices, with $U_l = |\hat{x}_l \hat{y}_l \hat{z}_l|$ and $U_r = |\hat{x}_r \hat{y}_r \hat{z}_r|$ respectively.

Let $\bar{c}_l$ and $\bar{c}_r$ be the centroids of the left and right point sets respectively. The corresponding translation t between the left and right coordinate systems can be estimated as

$$t = \bar{c}_l - R\bar{c}_r.$$

Let two sets of measured data points in collections of $\mathcal{M}$ and $\mathcal{M}'$ be $\mathcal{M} = \{v_{l,i}, i = 1, ..., m\}$ and $\mathcal{M}' = \{v_{r,i}, i = 1, ..., m\}$. Let $\hat{\mathcal{M}} = \{\hat{v}_{r,i}, i = 1, ..., m\}$ be the transformed data point collection from the left to the right coordinate system by applying rotation R and translation t. After obtaining the rotation and translation estimates, for a point $v_{l,i} \in \mathbb{R}^3$ in the left coordinate system $\mathcal{M}$, the mapping from the left to the right can be performed by:

$$\hat{v}_{r,i} = Rv_{l,i} + t,$$

where $\hat{v}_{r,i}$ is in $\hat{\mathcal{M}}$, the image of a left point $v_{l,i}$ in the right coordinate system.

3.4.2 Critical Point Based Approach

Liu and Rodrigues proposed a critical point based approach to obtain the rotation matrix R and translation vector t from three point correspondences [92].

Let us define correspondence vectors k_i by $k_i = v_{r,i} - v_{l,i}$ for $i = 1, 2, 3$. Given three non-collinear correspondence vectors k_i from $v_{l,i}$ to $v_{r,i}$, where $i = 1, 2, 3$, the rotation axis $g = (g_1, g_2, g_3)^T$ may be obtained by

$$g = \frac{(k_2 - k_1) \times (k_3 - k_1)}{\|(k_2 - k_1) \times (k_3 - k_1)\|}.$$

Let us define a correspondence pair as a pair of a left point $v_{l,i}$ corresponding to a right point $v_{r,i}$. Let $\underline{v_{l,i}}$ and $\underline{v_{r,i}}$ be the projections of a correspondence pair $(v_{l,i}, v_{r,i})$ on a plane perpendicular to the rotation axis g. After the rotation axis is determined, $\underline{v_{l,i}}$ and $\underline{v_{r,i}}$ may be obtained by

$$\underline{v_{l,i}} = (I - gg^T)v_{l,i},$$
$$\underline{v_{r,i}} = (I - gg^T)v_{r,i}.$$

Let $\underline{k_1}$ and $\underline{k_2}$ be the projections of the correspondence vectors k_1 and k_2 on the plane. For $\underline{k_1}$ and $\underline{k_2}$, if they are non-parallel, then the critical point $e \in \mathbb{R}^3$ may be determined by

$$e = \frac{\underline{v_{l,2}}^T \underline{v_{l,2}} - \underline{v_{r,2}}^T \underline{v_{r,2}}}{2\underline{k_2}^T G \underline{k_1}} G\underline{k_1} + \frac{\underline{v_{l,1}}^T \underline{v_{l,1}} - \underline{v_{r,1}}^T \underline{v_{r,1}}}{2\underline{k_1}^T G \underline{k_2}} G\underline{k_2}, \tag{3.6}$$

where

$$G = \begin{bmatrix} 0 & -g_3 & g_2 \\ g_3 & 0 & -g_1 \\ -g_2 & g_1 & 0 \end{bmatrix}.$$

From a projected correspondence $(\underline{v_{l,i}}, \underline{v_{r,i}})$, and the critical point e obtained by equation (3.6), we may determine the rotation angle θ as

$$cos\theta = \frac{(\underline{v_{l,i}} - e)^T (\underline{v_{r,i}} - e)}{(\underline{v_{l,i}} - e)^T (\underline{v_{l,i}} - e)}.$$

Then the rotation matrix R may be calculated by

$$R = I - Gsin\theta + (1 - cos\theta)G^2.$$

The translation vector t may be estimated from

$$t = (I - R^T)e + gg^T(v_{l,i} - v_{r,i}).$$

After obtaining the rotation matrix R and the translation vector t, the left data set $\mathcal{M} = \{v_{l,i}\}$ may be transformed to the right $\hat{\mathcal{M}} = \{\hat{v}_{r,i}\}$ by

$$\hat{v}_{r,l} = R(v_{l,i} - t).$$

3.5 The Hippocampus as an Object of Study for Alzheimer's Disease

In this section, we give an introduction on the hippocampus and MRI.

3.5.1 The Hippocampus

The hippocampus is a specialised cortical area in the human brain rolled into the medial temporal lobe [112]. It plays a critical role in long term memory and spatial navigation. It has three anatomical subdivisions: the dentate gyrus – in cross section, one of two interlocking C-shaped strips of cortex; the hippocampus proper – afferents from the dentate gyrus and septal nuclei, and efferents to the subiculum and septal nuclei; the subiculum – a transitional zone between the hippocampus proper and the entorhinal cortex.

The hippocampus is a key neuroanatomical structure for investigating ageing, Alzheimer's disease, schizophrenia and severe depression [56]. A hippocampus is shown in Figure 3.3.

3.5.2 MRI

Magnetic Resonance Imaging (MRI) is a medical imaging technique using radiology to visualise the internal structure and functions of the human body [52]. Since MRI can provide much greater contrast between the different soft tissues of the human body than computed tomography (CT), it becomes especially useful in neurological imaging. It is different from CT, because it uses no ionising radiation, but utilises a powerful magnetic field to align the nuclear magnetisation of hydrogen atoms in water in the human brain. Radio frequency fields are used to systematically change the alignment of this magnetisation, making hydrogen nuclei produce a rotating magnetic field detectable by the scanner. This signal is further processed to build up enough information to construct an image of the human brain. A MRI machine is shown in Figure 3.4.

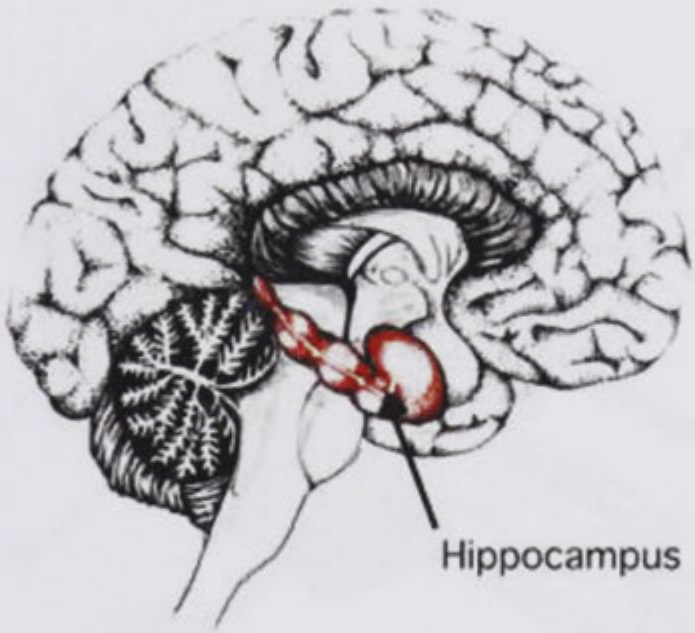


Figure 3.3: A hippocampus. The figure is taken from the Cognitive Control and Development Laboratory at UC Berkeley.

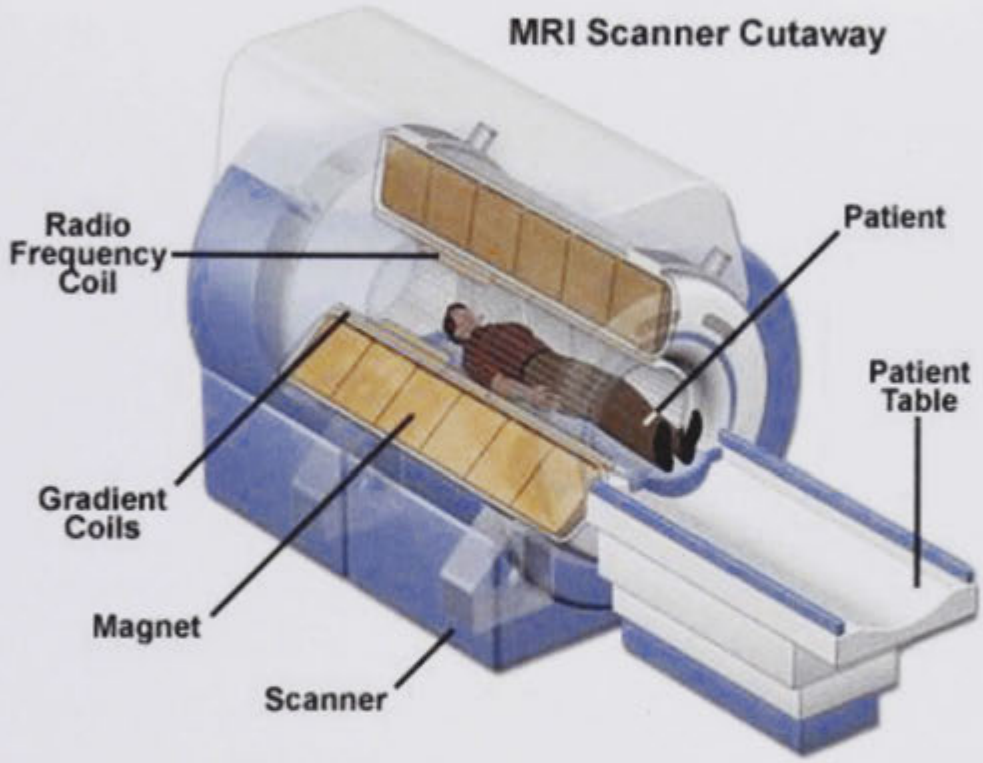


Figure 3.4: A MRI scanner. The figure is taken from the National High Magnetic Field Laboratory, USA.

3.6 Summary

In this chapter, we have reviewed the background material on Markov random fields and belief propagation. We have also provided background on curvature, scale space, 3D rigid body transformation estimation techniques based on three pairs of corresponding points, the hippocampus and MRI. In the next chapter, we will investigate a theoretical model for 3D shape matching based on Markov random fields.

Chapter 4

Theoretical Model for 3D Shape Matching Based on Markov Random Fields

In this chapter, we investigate a theoretical model for 3D shape matching based on Markov random field techniques. In order to incorporate both geometric and topological information of the surface, we explore the use of Markov random field techniques to perform correspondence matching. For correspondence matching, the quality of the matching feature directly influences the matching results. In this thesis, we use Gaussian curvature as a similarity measure criterion between the nodes because of its intrinsic geometric property. In addition, since it is difficult to carry out the correspondence matching in the original surface space, we perform a projection from the original space to the Markov random field space. We adopt a similar approach as proposed in [29]. The projection carries both geometric and topological information in terms of Gaussian curvature and mesh neighbourhood from the original space to the random field space. The Gaussian curvature is projected to the nodes of the MRF, and the mesh neighbourhood structure is projected to the edges. In this chapter, we present projection to MRFs and correspondence matching, and then give the probability measure and the Gibbs-Markov equivalence, and finally, define the energy function for correspondence matching.

4.1 Projection to MRFs and Correspondence Matching

Given two meshed surfaces M and M', if a map $f : M \rightarrow M'$ is locally isometric, then according to Gauss's Theorema Egregium, its Gaussian curvature K is preserved; that is, $K(p) = K(f(p))$, $p \in M$. By taking Gaussian curvature as a similarity measure criterion and the mesh neighbourhood structure as a compatibility measure criterion, the correspondence matching problem can be formulated as an optimisation of a function whose minimum corresponds to the best match between the two surfaces.

We do not perform the correspondence matching between M and M' directly, since it is not easy to capture both geometric and topological information. For this purpose we use Markov random field techniques: (1) the geometric information is incorporated as an attribute of the random variables; (2) and the topological neighbourhood information will be the MRF neighbourhood structure. Therefore, we will perform a projection from the original surface space to a Markov random field first.

Let M and M' be two meshed surfaces, and X be an MRF with $m = |M|$ nodes each with a state space of size $n = |M'|$. We perform a one-to-one mapping $g : M \rightarrow X$ which preserves Gaussian curvature at nodes and mesh topological structure at edges in X: (1) $K(g(i)) = K(i)$; and (2) if $i, j \in M$ are neighbours, then $g(i)$ and $g(j)$ are also neighbours. The correspondence matching problem between M and M' is posed as an energy function minimisation problem in X, and a point correspondence between M and M' is a realization of the MRF X in M'.

The projection carries both geometric and topological information in terms of Gaussian curvature and mesh neighbourhood from the original space to the projected random field space. After the projection, the correspondence between the two surfaces M and M' will be found by solving an energy function minimisation problem.

The energy function for the correspondence matching is defined in Section 4.3.

4.2 Probability Measure and Gibbs-Markov Equivalence

For ease of exposition, we introduce the following notation: given $i \in \{1, ..., m\}$, we associate node $i \in M$ to the ith variable in X, denoted by X_i, and to x_i, an arbitrary realisation of X_i in the state space $S = M'$. For a node $i \in \{1, ..., m\}$ of M, let $N(i)$ denote the neighbours of i in M. Let us define a neighbourhood system on X to be $N_X = \{N_X(i)\}_{i \in M}$ where $\{X_j | j$ is a neighbour of i in $X\}$. For simplicity, we identify $N_X(i)$ with $N(i)$; the meaning of $N(i)$ should be clear from the context. The vector $X = (X_1, ..., X_m)$ takes values in $\Omega = S^m = (M')^m$, the set of all m-tuples $x = (x_1, ..., x_m)$ with $x_i \in S$ for each i. We call Ω the configuration space.

The probability defined on the Markov random field can be obtained based on the energy function according to the Hammersley-Clifford Theorem upon satisfying certain conditions.

Assume the random field X is Markovian, then we have, by definition,

$$P(X_i = x_i | X_{M \setminus i} = x_{M \setminus i}) = P(X_i = x_i | X_{N(i)} = x_{N(i)}),$$

where $X_{M \setminus i}$ denotes all the random variables in M except X_i; $N(i)$ denotes the neighbours of node i.

For probability measures P on X, they are represented by $P = \{P(x)\}_{x \in \Omega}$ fulfilling the conditions $P(x) \geq 0$ and $\sum_{x \in \Omega} P(x) = 1$. The positivity condition means that for each configuration, its probability is non-negative.

According to the Hammersley Clifford Theorem as stated in Section 3.1.1 the probability on Markov random field is measured with Gibbs distribution. Gibbs distributions with an energy deriving from a Gibbs potential relative to a neighbourhood system are distributions of Markov fields relative to the same neighbourhood system [25].

4.3 Energy Function for Correspondence

After projecting the surface of a mesh model in $\mathbb{R}^3$ to a Markov random field, the projected space is represented as a graph. The set of edges describes the adjacency of the mesh nodes.

If we assume that the Markov property is based on pairwise cliques, the energy function $H(x)$ for the Markov random field X can be obtained based on two components

$$H(x) = w_1 H_1(x) + w_2 H_2(x), x \in \Omega, \tag{4.1}$$

where w_1 and w_2 are weights corresponding to the external and internal energy respectively on the Markov random field.

The first component is based on the curvature difference (Gaussian curvature) defined on singletons and corresponds to the external energy of the random field

$$H_1(x) = \sum_{c \in C_1} h_1(x_c) = \sum_i d_m(K(i), K(x_i)), i \in M, \tag{4.2}$$

where h_1 is the similarity measure of Gaussian curvature; d_m is the Minkowski distance defined as the L_p norm, $d_m(v, w) = L_p(v - w)$; and $K(i)$ denotes the Gaussian curvature of node i, $i \in M$ or $i \in M'$.

The second component of the energy function $H(x)$ incorporates the spatial restrictions based on the pairwise distance defined on 2-element cliques; it corresponds to the internal energy on the random field

$$H_2(x) = \sum_{c \in C_2} h_2(x_c) = \sum_{(i,j)} d_p(x_i, x_j), i \in M, j \in N(i), \tag{4.3}$$

where h_2 is the pairwise distance defined on 2-element cliques; (i, j) is a 2-element clique in X; and d_p is the pairwise distance defined on the meshed surface M'. Pairwise distances defined on the meshed surfaces M and M' help preserve the topological information of the shape.

Based on the probability measure defined in Equation (3.2) and the energy function defined above, we develop the following lemma about point correspondence matching.

Lemma 4.3.1. *Given an energy function $H(x)$ defined on the configuration space Ω, the point*

correspondence matching is formulated as the following optimisation problem

$$\arg \max_{x \in \Omega} P(x) = \arg \min_{x \in \Omega} H(x), \tag{4.4}$$

where $P(x)$is defined as in (3.2), and $H(x)$ is given as in (4.1).

The lemma follows since function $e^{-(H(x)/T)}$ is a monotonic decreasing function of $H(x)$.

The function $H(x)$ as in (4.1) based on the curvature difference d_m and pairwise distance d_p forms a Gibbs potential with regard to the neighbourhood system N_X, and the potential function forms a Gibbs distribution as in (3.2).

The Gibbs potentials are local in nature. To obtain an optimal solution to the probability distribution, the search space Ω could be very large, and in most cases it is intractable. Belief propagation is used in this thesis to deal with the optimisation problem with large state spaces.

The output of the MRF matching between M and M' is many-to-one. Since the neighbourhood constraint based on the pairwise distance is incorporated by using H_2 while defining the energy function, the near one-to-one correspondence output is expected for the matching. Our experiments show that the matching result is close to one-to-one. It is demonstrated that such a correspondence result is used effectively to achieve good 3D rigid body registrations.

4.4 Summary

In this chapter, we have presented projection to MRFs and correspondence matching. We have also given the probability measure and the Gibbs-Markov equivalence, and defined the energy function for the correspondence matching. The correspondence matching is obtained by minimising the energy function. However, the minimisation of the energy function is NP hard. A typical way to handle the issue is to approximate the global solution using techniques such as simulated annealing, graph cuts, and belief propagation (BP). We will explore the use of simulated annealing to minimise the energy function in Chapter 6. Among the approximation techniques, belief propagation is one of the most actively and widely investigated approaches. Belief propagation has proven to be successful in many applications including early vision [38,

134, 137]. In the next chapter, we will investigate how to use belief propagation to minimise the energy function defined in this chapter.

Chapter 5

Belief Propagation

5.1 Introduction

We present the key algorithmic approaches of this thesis in this chapter. According to Chapter 4, the best match between two meshed 3D surfaces is obtained by solving the correspondence matching problem between the nodes of the two meshed surfaces. This correspondence matching is formulated as an energy function minimisation problem by using MRF techniques.

To minimise such an energy function is NP hard, since the computational complexity increases exponentially with the size of the state space at each node. For a moderate size of the state space, the search space could be large for an exhaustive search. In this chapter we investigate applying belief propagation to solve the probabilistic inference problem arising from minimising the energy function for 3D shape matching. We present an efficient belief propagation algorithm incorporating a sparse update approach adapted to 3D shape matching, which is much faster than the standard BP for large state spaces. We compare the derived sum and max product algorithms for shape matching in 3D. We then discuss some implementation issues. We give the algorithm's computational complexity, and discuss its convergence properties. We also describe another key contribution of the thesis, a novel clamping technique, which combines the belief propagation message updating algorithm with an external process that gives feedback information about matching which the BP uses to clamp probabilities of sets of nodes.

5.2 Energy Function Definition

The Gibbs distribution corresponding to a Markov random field X is written as follows according to Equation (3.2)

$$P(x) = Z^{-1}exp(-H(x)), \tag{5.1}$$

where x is a configuration of the MRF X, $Z = \sum_x exp(-H(x))$ is the partition function, $H(x)$ is the energy function.

Recall from Equation (4.2) and (4.3) that the energy function $H(x)$ is obtained based on the following two components:

$$H_1(x) = \sum_{c \in C_1} h_1(x_c) = \sum_{i \in C_1} d_m(K(i), K(x_i)), i \in M,$$

and

$$H_2(x) = \sum_{c \in C_2} h_2(x_c) = \sum_{(i,j) \in C_2} d_p(x_i, x_j), i \in M, j \in N(i),$$

where C_1 is the set of singletons, and d_m is the Minkowski distance; C_2 is the set of 2-element cliques, and d_p is the pairwise distance defined on the meshed surface M'.

The total energy $H(x)$ according to Equation (4.1) is

$$H(x) = H_1(x) + w * H_2(x), \tag{5.2}$$

where w is the balancing factor between internal energy and external energy. Therefore, (5.1) is rewritten as

$$\begin{aligned} P(x) &= Z^{-1}exp(-(\sum_{c \in C_1} h_1(x_c) + \sum_{c \in C_2} h_2(x_c))) \\ &= Z^{-1}exp(-(\sum_{i} d_m(K(i), K(x_i)) + \sum_{(i,j)} d_p(x_i, x_j))), \end{aligned} \tag{5.3}$$

where $i \in M$ and $j \in N(i)$.

In the following section, we will formulate the energy function minimisation problem for

3D shape matching under a probabilistic inference framework using belief propagation.

5.3 Sparse Update Belief Propagation

Sparse update belief propagation is an efficient way to perform the belief propagation. We developed sparse update belief propagation based on standard belief propagation formulation. There are two algorithms associated with belief propagation: sum-product and max-product algorithms.

5.3.1 Belief Propagation Formulation

Equation (5.1) gives an exact probability distribution on the Markov random field. As shown by Boykov et al. (2001) in [22], and Kolmogorov and Zabih (2004) in [74], to calculate the exact marginals or Maximum a posteriori on the MRF is intractable. However, if we are able to formulate the problem under a probabilistic inference framework, we may use belief propagation to approximate the solution of the energy function minimisation problem.

Set

$$\phi_i(x_i) = exp(-d_m(K(i), K(x_i))), i \in M, \tag{5.4}$$

and

$$\psi_{ij}(x_i, x_j) = exp(-d_p(x_i, x_j)), i \in M, j \in N(i). \tag{5.5}$$

We say that $\phi_i(x_i)$ is the similarity measure, and $\psi(x_i, x_j)$ is the compatibility measure. Therefore, we rewrite (5.3) as [155]

$$P(x) = Z^{-1} \prod_{(ij)} \psi_{ij}(x_i, x_j) \prod_{i} \phi_i(x_i). \tag{5.6}$$

The belief $b_i(x_i)$ is defined as the probability of node i being in state x_i and represents the marginal probability of x_i; $m_{ij}(x_j)$ is defined as the message from node i to node j about what state node j should be in, and j is a neighbour of i. The belief b_i at a node i is proportional to

the product of the local evidence at that node and all the messages coming into that node [155]

$$b_i(x_i) = z\phi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i) \tag{5.7}$$

where z is a normalisation constant.

The messages between nodes are updated as follows [155]

$$m_{ij}(x_j) = \sum_{x_i} \Big[\phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i) \Big]. \tag{5.8}$$

Equation (5.8) is the basis of the sum-product algorithm. We note that on the right-hand side, we take the product of all the messages coming into node i, except the one coming from node j. Further, the summation ranges over all the states x_i the node i may find itself in. In a tree structured MRF, after message passing from leaf nodes towards a single root, the exact marginals $b_i(x_i)$ for the entire joint distribution can be obtained. However, for an MRF with loops, it is not guaranteed that the belief propagation will converge or produce an approximate solution close to the global optimal.

5.3.2 Sparse Update Loopy Belief Propagation for 3D Shape Matching

When matching two mesh models representing the shape of objects, there is a topological structure which ensures that certain combinations of point matches cannot occur. For example, given a match between a node X_i in X and a node x_i in the other mesh model M', an immediate neighbour X_j of X_i cannot match a point x_j that is very far from x_i in M'. In other words, one cannot match a point on one hippocampus to the head of the other hippocampus while matching its immediate neighbour to the tail of the other. To embody this we modify standard belief propagation as follows.

We define the pairwise potential function $\psi_{ij}(x_i, x_j)$ as a step function

$$\psi_{ij}(x_i, x_j) = \begin{cases} 1, & \text{if } x_j \in N(x_i) \\ 0, & \text{otherwise,} \end{cases} \tag{5.9}$$

where $N(x_i)$ is the set of neighbours of x_i in M'.

Equation (5.9) corresponds to the two extreme cases of Equation (5.5), in which 1 corresponds to the case when $d_p(x_i, x_j)$ equals to zero and 0 corresponds to the case when $d_p(x_i, x_j)$ is infinity. Therefore, the definition of Equation (5.9) is not detrimental to the theoretical guarantees of MRFs. Detail discussion on the convergence of BP is presented in Section 5.5.2.2.

By using this property, Equation (5.8) is changed to

$$m_{ij}(x_j) = \sum_{x_i \in N(x_j)} \Big[\phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i) \Big]. \tag{5.10}$$

If the compatibility measure over pairwise nodes i and j $\psi_{ij}(x_i, x_j)$ vanishes, then the summation operators for the message updating rule for the sum product algorithm is performed over the non-vanishing neighbours only. Therefore, (5.10) is equivalent to (5.8) when defining ψ_{ij} as in (5.9).

Equation (5.10) is the sum-product message updating equation. Given Equation (5.9), when a message is passed from node i to node j, this message has a positive value if x_i is in the neighbourhood of x_j, and zero otherwise. In Equation (5.9), any match outside the immediate neighbourhood has zero probability; such a cut-off is beneficial to computation speed. When the set $N(x_i)$ is more generally defined as $N(x_i) : \{x_j | d_p(x_i, x_j) \leq d\}$ for given d, the computation time then increases exponentially with d.

The complexity analysis is discussed in the Section 5.5. The correspondingly derived sum and max product algorithms for 3D shape matching are formulated in the following two subsections.

5.3.3 Sum-Product Algorithm

The sum-product algorithm takes the product of all incoming messages and then takes the sum over the products. By computing the marginal distributions of each node, the algorithm obtains the maximum marginalization estimation for the whole graph. In our implementation, messages are updated sequentially. For numerical stability reasons, we initialise all messages to a constant value of 1.

Let $m_{ij}^{(t)}$ be the message that node i sends to a neighbouring node j at time t. The algorithm for sum-product is given as follows in Algorithm 2.

Algorithm 2 Sum-Product

(1) Initialise all the messages $m_{ij}^{(0)}(x_j)$ to 1;
(2) Iterate over time t
– Update all messages sequentially according to

$$m_{ij}^{(t)}(x_j) \leftarrow \sum_{x_i \in N(x_j)} \left[\phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}^{(t-1)}(x_i) \right]$$

– Compute beliefs from

$$b_i^{(t)}(x_i) \leftarrow z\phi_i(x_i) \prod_{j \in N(i)} m_{ji}^{(t)}(x_i)$$

– Get the best estimate of the labelling for each node from

$$\hat{x}_i = \arg\max_{x_i} b_i(x_i).$$

– Check whether the iteration number is greater than the pre-defined maximum iteration number. If yes, terminate; otherwise, continue the loop.

The maximum iteration number is set empirically in Algorithm 2.

5.3.4 Max-Product Algorithm

Another popular formulation of the message update step is as a max-product. By replacing the summation in (5.10) with a maximisation, we get the max-product algorithm:

$$m_{ij}(x_j) = \max_{x_i \in N(x_j)} \left[\phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i) \right]. \tag{5.11}$$

The max-product algorithm computes the Maximum A Posteriori (MAP) estimate of the whole MRF. For a given state and its neighbourhood, max product will give a high score if at least one state in the neighbourhood has high probability, whereas sum product only gives a high score if the average over the neighbours is high. We will see in Section 5.5.2.2 that max product performs better than sum product in our 3D shape matching results.

5.4 Implementation

5.4.1 Message Normalisation

Since the local evidence measure $\phi_i(x_i)$ over a node i is defined as $exp(-d_m(K(i), K(x_i)))$, its range is $(0, 1]$. The range of the compatibility measure $\psi_{ij}(x_i, x_j)$ of Equation (5.9) is $\{0, 1\}$. Therefore, if we directly perform message updating and belief calculating rules, after only a few iterations, an underflow problem will occur. To avoid this problem, we used a normalisation technique which handles the problem effectively.

The main idea of the normalisation is to use the monotonic property of the log function over a variable. Since our aim is to get the maximum marginalization and Maximum a posteriori estimates, a log transformation will not change the arg max and arg min of a function.

The normalisation algorithm is shown as follows:

Algorithm 3 Message Normalisation

(1) loop through the messages:
– search for the maximum of all the messages;
(2) Divide each message by the maximum;
(3) Add 1.0 to all the messages, so that $m_{ij}(x_j) \geq 1.0$;
(4) Take the log values of all the changed messages;
(5) Add 1.0 to the log of all messages, so that $m_{ij}(x_j) \geq 1.0$;
(6) New value of a message is $log(\frac{m_{ij}(x_j)}{max} + 1.0) + 1.0$.

The main purpose of Algorithm 3 is to transform the messages to values above 1.0 and use log transformation [2] to avoid the underflow problem. We transform the messages to values above 1.0 to make sure that the messages after applying log transformation are still positive. Typical data transformation methods for normalisation include adding constants, raising to a power, using logarithm, taking square root, and applying trigonometric transformation. We choose to use the methods of adding constants and applying log transformation because we need to maintain the monotonic property while performing the normalisation. Similar techniques have been used to compute posterior probabilities in [150].

5.4.2 Storage Consideration

To save space, we use an adjacency list to represent the neighbourhood structure rather than an adjacency matrix.

5.5 Complexity and Convergence Analysis

5.5.1 Complexity Analysis

The computational complexity of the standard sum and max product belief propagation algorithms is $O(kmn^2)$, where k is the number of iterations, m is the size of MRF, and n is the number of states at each node. In our application, the two meshes generally have a similar number of nodes, as human hippocampi are of similar size, so $m \sim n$. Thus, since the number of possible assignments for each node –the state space– is the number of nodes of the other mesh, the order is approximately km^3.

Since ψ_{ij} is defined as a step function, the domain for the summation and maximisation operators changes from the whole state space to the current state's neighbours only, the size of the search space reduces from m to $\bar{l}$, where $\bar{l}$ is the mean number of neighbours of a node in M'. Therefore, the computational complexity for the modified sum and max product message updating rules reduces to $O(km^2\bar{l})$.

As the number of neighbours l for a particular node is generally small, typically varying from 3 to 9, compared to the number of states m which ranges from hundreds to thousands, the computational complexity is significantly reduced. In our experiments, the time for each iteration is reduced from 1 minute for the standard BP to 2 seconds for the modified BP for a synthetic data set with two hundred nodes, which is 30 times faster; and reduced from 16 hours to 6 minutes for a real hippocampus data set with two thousand nodes, which is 160 times faster.

The beliefs obtained by performing message update by using Equation (5.10) and by using Equation (5.8) are the same. We have verified this by performing an experiment comparing the running information of belief propagation between using full state and restricting to a node's

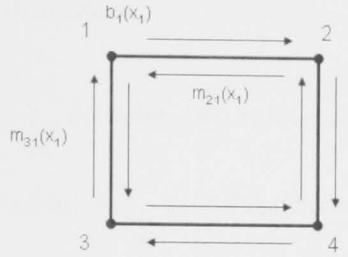


Figure 5.1: Loopy graph with four nodes.

immediate neighbours only in the message updating rule while performing the matching. The experiment shows that the messages, the beliefs, and the most likely state at every node for each iteration are the same. This illustrates that in terms accuracy, the sparse update BP is the same as the standard BP.

5.5.2 Comments about Convergence

5.5.2.1 The Problem of Loopy Graphs

In this section, we give an intuitive overview of the BP process. During the message updating process, there is a self referring problem in a loopy graph. Take an example of a loopy graph with four nodes as shown in Figure 5.1.

To calculate belief at node 1, we need to get messages from all its neighbours, $m_{21}(x_1)$ and $m_{31}(x_1)$,

$$b_1(x_1) = z\phi_1(x_1)m_{21}(x_1)m_{31}(x_1).$$

Furthermore, $m_{21}(x_1)$ is calculated as,

$$\begin{aligned}
\overbrace{m_{21}(x_1)} &= \sum_{x_2} \phi_2(x_2)\psi_{12}(x_1, x_2)m_{42}(x_2) \\
m_{42}(x_2) &= \sum_{x_4} \phi_4(x_4)\psi_{24}(x_2, x_4)m_{34}(x_4) \\
m_{34}(x_4) &= \sum_{x_3} \phi_3(x_3)\psi_{34}(x_3, x_4)m_{13}(x_3) \\
m_{13}(x_3) &= \sum_{x_1} \phi_1(x_1)\psi_{13}(x_1, x_3)\overbrace{m_{21}(x_1)}.
\end{aligned}$$

Eventually, $m_{21}(x_1)$ refers to itself while the messages propagate over the network. Similarly for $m_{31}(x_1)$, it refers to itself after the message propagates across three nodes.

$$\begin{aligned}
\overbrace{m_{31}(x_1)} &= \sum_{x_3} \phi_3(x_3)\psi_{13}(x_1, x_3)m_{43}(x_3) \\
m_{43}(x_3) &= \sum_{x_4} \phi_4(x_4)\psi_{34}(x_3, x_4)m_{24}(x_4) \\
m_{24}(x_4) &= \sum_{x_2} \phi_2(x_2)\psi_{24}(x_2, x_4)m_{12}(x_2) \\
m_{12}(x_2) &= \sum_{x_1} \phi_1(x_1)\psi_{12}(x_1, x_2)\overbrace{m_{31}(x_1)}.
\end{aligned}$$

From the above observation, we can conclude that the messages in a loopy graph refer to themselves after propagating over certain nodes during the message updating process. Since the message is self-referring, the message updating process can not obtain more information from the initial values of the messages. In a loopy graph, the end state is independent of the initial values as opposed to singly connected graphs. The similarity measure and compatibility measure play an important role during the message updating process. The intuition behind loopy belief is that by initialising the messages to non-negative values, after a certain number of iterations, the messages propagate over the network, and the marginals at each node will gradually converge to their correct values from the similarity measure over singletons and the compatibility measure over pairwise nodes.

5.5.2.2 Max versus Sum Product

The loopy belief propagation in this 3D shape matching problem is not guaranteed to converge. In this subsection, we use current theoretical results in this area to do some analysis of our problem.

The marginal probability of x_i is $b_i(x_i)$, and the marginal distribution is $P(x_i)$. The MAP of x_i is obtained by

$$x_i^* = \arg \max_{x_i} b_i(x_i).$$

The sum product algorithm provides a method to compute the marginal distributions $P(x_i)$ for a single node i over a graph. For a graph without cycles, the sum product can be used to perform exact marginalization over the graph. It has been shown by Pearl (1988) [114] that if applying the sum product to a singly-connected Bayesian net, it is guaranteed that the beliefs at each node will converge to the correct posterior probabilities. Equivalently, for a tree-structured graph, the sum product of belief propagation computes the marginals for each single variable exactly. In this case, the message updating process (5.8) converges in a finite number of iterations, and the iteration number for beliefs to reach a steady state is bounded by the longest path in the graph. After convergence, the belief (5.7) equals the correct marginal $P(x_i)$.

However, for graphs with cycles, running belief propagation is not guaranteed to converge theoretically [154]. The messages updated by (5.8) are not guaranteed to converge to a fixed point after certain number of iterations.

According to theoretical results about probabilistic inference on a graph with loops by Weiss et al. (2001) [149], for a graph with a single loop, if the max product converges, the resulting MAP configuration has higher probability than any other configuration in the graph. By running sum product, BP produces correct marginal estimates; and by running max product, BP produces correct MAP estimates.

Furthermore, according to Weiss and Freeman (1999) [148], if a graphical model is Gaussian distributed, then when belief propagation converges, it gives the correct posterior estimate for any graph topology. It has also been shown that even for non-Gaussian distributed graphi-

cal models, the convergence values for a loopy graph after running the max-product algorithm are at least local maxima of the posterior probability [148].

Since our potential function over pairwise cliques is defined as a step function in (5.9), the MRF is not strictly distributed as a Gaussian. The step function definition over the pairwise cliques sharpens the edge, as opposite to a pure Gaussian kernel.

Empirically, Weiss shows a simulation result in [147], which demonstrates that even for non-Gaussian distributed graphical models max product BP produces correct MAP estimates. Our experimental results (see Chapter 8) confirm that the matching results by our max product algorithm are better than by the corresponding sum product counterpart, which supports the above theoretical results and is consistent with Weiss's simulation result.

5.6 Clamping Techniques

In matching experiments, we found there are many mis-matches among the correspondence matches. From the 3D rigid body registration results from Chapter 8, we know the matching rate following loopy belief propagation is 15% to 20%. As to achieve dense correspondences we would like a correct match for the majority of nodes, the rate of correct matchings is low for this purpose. In order to increase the rate of correct matchings, we propose a clamping technique. The intuition behind the clamping technique is that if we know that a particular state of a given node has high probability from an external process while running the loopy belief propagation message updating rule, then we can clamp that state for that node (set and fix its probability to a high value) and allow other nodes to go through the normal message updating procedure. In this way, the rate of correct matchings can be increased as the correctly clamped states propagate their increased probabilities. We iteratively run the message updating process and the external process until no progress is made.

Definition 1. *We define clamping as restricting the set of possible state assignments of a particular node based on information that is available externally to the standard belief propagation process.*

Let W be a process that is external to the standard belief propagation process. For a node

i, we set its possible state assignment as a set of states S_i. S_i is obtained from W according to which match is a good match. Let $P_i(S_i)$ be the probability that node i is in state S_i. If node i is clamped, then $P_i(S_i)$ is increased. Otherwise, it is decreased.

Some external process has identified a set of nodes for which some state assignments are more likely, or other state assignments are not possible. The aim is that the neighbours of these nodes are attracted to states that are consistent with these fixed points in the probability distribution over state assignments.

5.6.1 Clamping in Practice

In our experiment, we choose 3D rigid body registration combined with a distance metric as the external process to inform loopy BP as to which states should be clamped. Clamping then clamps the correct state using the feedback from 3D rigid body registration while performing the loopy belief propagation.

As shown in Figure 5.2, let us define two neighbouring nodes i and j. i is sending message to j. i is in state t, j is in state s and t is s's neighbour. Let $map(i)$ be the correspondence obtained for node i in BP. Inliers are those points i for which $d(i, map(i))$ is less than some threshold. If i is an inlier, then i is clamped to $map(i)$. The clamping process checks whether t is equal to $map(i)$:

If yes, the program goes as usual to calculate the message value.

If no, the calculated message is set to zero. This means that this state assignment to t is not preferred. This corresponds to a neighbouring case for the following belief propagation message updating rule equation

$$m_{i \to j}(s) \leftarrow \max_{t \in \{N(s) \cap map(i)\}} \left[\phi_i(t) \psi_{ij}(s,t) \prod_{k \in N(i) \setminus j} m_{k \to i}(t) \right].$$

In this case, in the above equation, if the node i is clamped to $map(i)$, and if t is not equal to $map(i)$, then $\phi_i(t)\psi_{ij}(s,t) \prod_{k \in N(i) \setminus j} m_{k \to i}(t)$ is set to zero.

Therefore, the inlier information from the 3D rigid body registration is integrated with the belief propagation message updating process. In this way, the neighbouring matching of inlier

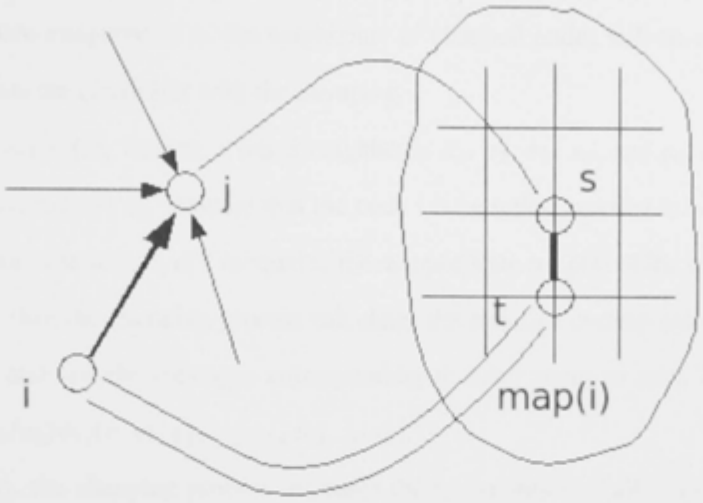


Figure 5.2: Clamping: integrating 3D rigid body transformation with Belief Propagation. i is in state t, j is in state s, and t is s's neighbour. If i is an inlier, then i is clamped to state $map(i)$. Belief propagation is sending message from node i to j.

information can be enhanced. By using clamping techniques, the rate of correct matchings can be significantly increased.

The clamping algorithm is shown in Algorithm 4.

Algorithm 4 Clamping algorithm

1 In message updating procedure, check every node, if the node is a good match, then it is clamped.
2 Traverse every node,
2.1 If it is not clamped, the program goes as usual.
2.2 If it is clamped, check whether t is equal to $map(i)$ of the clamped node i.
2.2.1 If yes, the program goes as usual.
2.2.2 If no, then the message $\phi_i(t)\psi_{ij}(s,t)\prod_{k\in N(i)\setminus j} m_{k\to i}(t)$ is set to zero.

In iterations of belief propagation that occur after clamping, the updated probability of the clamped nodes will impact the neighbours. The neighbours will receive messages that increase the likelihood of their being in a state that is consistent with the mapped state of the clamped node. That is, for states that are consistent with $map(i)$, the message will give a non-zero probability. On the other hand, for state assignments that are not consistent with $map(i)$,

the messages about these states will be zero. The impact of this propagation will be that, in general, the state assignments of the neighbours of clamped nodes will be updated to favour assignments that are consistent with the clamping.

Taking an example, assume s has 5 neighbours s_1, s_2, s_3, s_4, and s_5, and then t has 5 states s_1, s_2, s_3, s_4, and s_5. Assume that the node i is an inlier, and that it is clamped to state $map(i)$. Suppose the state s_3 of t is equal to the mapped state $map(i)$ of the node i, i.e. $N(s) \cap map(i) = s_3$, then the clamping process calculates the message corresponding to the state s_3 of t as usual, and sets the messages corresponding to other states to zero, $\{0, 0, 0, p_3, 0, 0\}$, where $p_3 = \phi_i(s_3)\psi_{ij}(s, s_3)\prod_{k \in N(i) \setminus j} m_{k \to i}(s_3)$.

In this way, the clamping process increases the probabilities of allowable states, and decreases the probabilities of non-allowable states. In the end, the rate of correct matchings is increased.

The experiments of matching with clamping for real hippocampus data are demonstrated in Chapter 8. The clamping technique increases the rate of correct matching from initial 15% to 20% up to 90% to 95%.

Now what we have realised is to incorporate the external process validation to the message update rule. Another consideration could be to incorporate the external process directly to ψ. However, this may require extra computational complexity, and also computational stability would be a problem. The main reason that we use clamping in our loopy belief propagation algorithms is the need to incorporate a global constraint to our local pairwise MRF framework. By using an external process such as 3D rigid body registration, which captures the global constraints between two hippocampi, we are able to identify inliers of the correspondence matches. The inliers are further integrated to the loopy belief propagation message update rule. The message update process of loopy belief propagation is local. In this way, we achieve the incorporation of a global constraint to our local MRF framework.

5.7 Summary

In this chapter, we have investigated belief propagation techniques to minimise the energy function for 3D shape matching by performing probabilistic inference on a loopy graph. We have proposed a sparse update belief propagation algorithm adapted to 3D shape matching to perform the probabilistic inference. The sparse update algorithm is much faster than the standard BP. The corresponding computational complexity and convergence properties have been discussed also. We have applied both sum and max product algorithms of belief propagation to minimise the energy function. The clamping technique is an effective approach to enhance the shape matching rate, and increase the inlier support set. In the next chapter, we will present randomised algorithms to minimise the energy function for 3D shape matching.

Chapter 6

Randomised Algorithms to Minimise the Energy Function for 3D Shape Matching

Given two meshed surfaces M and M', 3D shape matching can be formulated as a match between two 2-dimensional surfaces embedded in $\mathbb{R}^3$ according to their shape similarity. As shown in Chapter 4, the matching between two surfaces can be established by finding node correspondence according to a similarity measure in terms of curvature and mesh neighbourhood structure.

Suppose Gaussian curvature measured by estimated principal curvatures [43] for each individual node of a meshed surface is given. According to Chapter 4, by using Gaussian curvature to be the similarity measure, the correspondence matching can be formulated as an optimisation problem based on an energy function defined on a Markov random field (MRF). MRF is used to preserve both geometric and topological information. The best match between two surfaces can be obtained by minimising the energy function. The minimisation is an NP hard problem. For matching with real data, the state space is quite large, which is typically more than 2000. We have addressed this large state space energy function minimisation problem with real data of the hippocampus by using belief propagation in Chapter 5. To demonstrate alternative methods of minimising the energy function to belief propagation, we explore Gibbs sampler and Metropolis sampler based simulated annealing algorithms. Although the Gibbs sampler and Metropolis sampler based simulated annealings work effectively with the syn-

thetic data having a state space of size 200, it does not work with the real data with a state space of size 2000. The Gibbs sampler and Metropolis sampler based simulated annealings have similar performance to belief propagation with small state space synthetic data. In this chapter, we investigate simulated annealing (SA) and a Las Vegas type algorithm as a means to solve the minimisation problem with synthetic data.

6.1 Simulated Annealing

In Section 2.4, we introduce global methods for energy function minimisation. The main reason that we choose Simulated Annealing (SA) to minimise the energy function is that we want to investigate how randomised algorithms are used to minimise the energy function for 3D shape matching. These algorithms have similar or greater complexities. What we found was that the algorithms were effective, but could not handle the large state space. We used SA as an alternative method for the experiments on synthetic data, and for real data we focus on belief propagation that is described in Chapter 5.

Simulated Annealing (SA) is an analogy to a physical phenomenon where molecules in a liquid move more freely at high temperatures than at low temperatures [93]. With a slow decrease in the temperature, the thermal mobility of the molecules is gradually reduced, and finally the molecules form a pure crystal which corresponds to a state with minimum energy.

When the annealing process is applied to a Markov random field X, at each step, a new configuration x' of the random field is constructed from the current configuration x. x is changed to x' by randomly changing a random variable X_i, where i is a randomly selected node, from x_i to x'_i. x_i and x'_i are the current and new values of the random variable X_i at the single node i.

We explored two ways for sampling during annealing: Gibbs sampler and Metropolis sampler. For sampling schemes, we performed both sequential and random visiting schemes for X. A sweep for the sequential visiting scheme is a sequential visit of all nodes [45]. A random visit for the random visiting scheme is to randomly visit a single node each time.

We associate the process of Gibbs sampler and Metropolis sampler based simulated an-

nealings with a global control parameter temperature T_n. At low temperatures the distributions concentrate on states that decrease the energy function, whereas at high temperatures the distribution is essentially uniform. T_n is initially large, so the process avoids becoming trapped in local minima. Then temperature is gradually lowered and the matching is iteratively adjusted to minimise the energy function. This gradual reduction of temperature simulates "annealing" and has been shown to converge to the global minima of the energy function [45, 102].

During the annealing process, while in very high temperature, each state has almost equal probability to be the current state. While in low temperature, only the state with lower energy would have higher probability to be the current state [152]. The cooling schedule we adopted is [124]

$$T_n = \frac{1}{A * log(B + n)}, \tag{6.1}$$

where T_n is temperature at time n, n is the annealing iteration number, and A and B are constants. As n increases, the temperature gradually reduces.

6.1.1 Gibbs Sampler

The Gibbs sampler was introduced in the context of image processing by Geman and Geman in 1984 [45]. A special case of the Metropolis-Hastings algorithm [152], the Gibbs sampler is applicable when the joint distribution for a random field is not known explicitly, but the conditional distribution of each variable is known. The basic idea is to generate an instance from the distribution of each variable in turn, conditional on the values of the other variables. The sequence of the samples comprises a Markov chain, and the stationary distribution of the Markov chain is the underlying implicit joint distribution.

The key to the Gibbs sampler is that one only considers univariate conditional distribution – the distribution when all of the random variables but one are assigned fixed values. Such conditional distributions are far easier to compute than complex joint distributions and usually have simple forms. Thus, one computes n random variables sequentially from the n univariate conditionals rather than generating a single n-dimensional vector in a single pass using the full joint distribution.

Consider a bivariate random variable (x, y), and suppose we wish to compute one or both marginals, $P(x)$ and $P(y)$. The idea behind the Gibbs sampler is that it is far easier to consider a sequence of conditional distributions, $P(x|y)$ and $P(y|x)$, than it is to obtain the marginal by integration of the joint density $P(x, y)$, e.g. $P(x) = \int P(x, y)dy$. The sampler starts with some initial value y_0 for y and obtains x_0 from the conditional distribution $P(x|y = y_0)$. The sampler then uses x_0 to generate a new value of y_1, drawing from the conditional distribution based on the value x_0, $P(y|x = x_0)$ [143].

Suppose we wish to sample from the joint distribution for a state variable $X = \{X_1, \cdots, X_n\}$ given by $P(x_1, \cdots, x_n)$, where the range of the X_i may be either continuous or discrete. The Gibbs sampler does this by repeatedly replacing each component with a value chosen from its distribution conditional on the current values of all other components [111].

Let us define $X^{(0)}, X^{(1)}, X^{(2)}, \cdots$ as the state variable X in time $0, 1, 2, \cdots$ respectively, and $X^{(t)}$ and $X^{(t-1)}$ as the state variable X in time t and $t-1$. The algorithm can be described as simulating a homogeneous Markov chain, $X^{(0)}, X^{(1)}, X^{(2)}, \cdots$. The homogeneous Markov chain means that the transition probabilities do not change over time (the probability of going from one state to another today is the same as it will be at any time in the future). Let us define $x_1^{(t-1)}, x_2^{(t-1)}, \cdots, x_n^{(t-1)}$ to be the values of random variables $X_1, X_2, \cdots, X_n$ at time $t-1$, and $X_1^{(t)}, X_2^{(t)}, \cdots, X_n^{(t)}$ to be the random variables $X_1, X_2, \cdots, X_n$ at time t. The procedure for generating $X^{(t)}$ from $X^{(t-1)}$ can be expressed as follows [111],

- Compute $X_1^{(t)}$ from the distribution for X_1 given $x_2^{(t-1)}, x_3^{(t-1)}, \cdots, x_n^{(t-1)}$.
- Compute $X_2^{(t)}$ from the distribution for X_2 given $x_1^{(t)}, x_3^{(t-1)}, \cdots, x_n^{(t-1)}$.
- $\vdots$
- Compute $X_i^{(t)}$ from the distribution for X_i given $x_1^{(t)}, \cdots, x_{i-1}^{(t)}, x_{i+1}^{(t-1)}, \cdots, x_n^{(t-1)}$
- $\vdots$
- Compute $X_n^{(t)}$ from the distribution for X_n given $x_1^{(t)}, x_2^{(t)}, \cdots, x_{n-1}^{(t)}$

Note that the new value for X_{i-1} is used immediately when choosing the next value for X_i.

The basic operation used in the Gibbs sampling algorithm is the generation of a random value for an element of a state, X_i, from its conditional distribution given the current values of all the other elements, X_j, for $j \neq i$. The speed of the algorithm depends crucially on whether this operation can be done quickly [111].

A periodic Gibbs sampler is used in our experiments. During sampling, the random variables are updated in a predefined order $X_1, X_2, \cdots, X_n$, where $\{X_i\}_{1 \leq i \leq n}$ is an enumeration of all the nodes of X.

For our random field based Gibbs sampler, the sampling procedure is formulated as follows. For distribution $P(x) = P(x_i, i \in M)$ where M is the index set of the random field from which we wish to sample, and given a random initial state of the Markov chain, the Gibbs sampler sequentially updates the state of one of the nodes by drawing from the distribution of the node conditioned on the states of the remaining nodes of the random field.

A sweep is one sequential visit to all nodes of the random field. As the number of sweeps tends to infinity, from the long run behaviour of the random field-valued Markov chain $\{X^{(t)}\}_{t \geq 0}$, its distribution approaches the Gibbs distribution [45].

The Gibbs sampling algorithm for our experiments is summarised as follows. The iteration number I in the below algorithm is set empirically according to experience.

Algorithm 5 Gibbs Sampling Algorithm

(1.) Randomly initialise $\{x_i, i \in M\}$.
while t is less than the iteration number I **do**
(2.1.) traverse space M,
(2.1.1.) sample $x_i^{(t+1)}$ according to $p(x_i | x_{M \setminus i}^{(t)})$, which is equivalent to $p(x_i | x_{N(i)}^{(t)})$ where $N(i)$ is the neighbours of i;
(2.1.2.) new values for x_j where $j < i$ are used straight away in subsequent sampling steps.
end while

6.1.1.1 Annealing with Gibbs Sampler

The theory behind annealing based on Gibbs sampling is that the joint distribution for the Markov random field is not known explicitly, but the conditional distribution of each variable is known. The assumption is that the knowledge of the conditional distribution is sufficient to

determine a joint distribution [45].

Since the MRF involves a large number of variables, direct specification of a joint distribution for it is difficult. However, if the set of full conditionals can be specified, then we can draw samples accordingly. We specify the full conditional probabilities by using the Markovian neighbourhood structure based on singletons and pairwise cliques.

We obtain the conditional probability for the state x_i at node i from the local characteristics of the MRF by

$$P(x_i|N(i)) = \frac{exp(-\frac{1}{T_n}[h_1(x_i) + w * \sum_{j \in N(i)} h_2(x_i, x_j)])}{\sum_{y_i \in S} exp(-\frac{1}{T_n}[h_1(y_i) + w * \sum_{j \in N(i)} h_2(y_i, y_j)])}, \tag{6.2}$$

where, $N(i)$ denotes the neighbours of i; T_n is the controlled temperature at loop n as defined in (6.1); and h_1 and h_2 are as defined in Chapter 4. w is the balancing factor between the similarity measure of Gaussian curvature and the pairwise distance. The local conditional distributions are dependent on the globally controlled temperature T_n. At low temperatures the local conditional distributions concentrate on states that decrease the energy function, whereas at high temperatures the distribution is essentially uniform. The limiting cases, $T_n = 0$ and $T_n = \infty$, correspond respectively to greedy algorithms such as gradient descent and undirected purely random changes [45].

The sampling procedure at a particular node is first to calculate the conditional probabilities for all the states of the node according to Equation (6.2). We then randomly generate a probability and check into which range of the calculated conditional probabilities it falls. The state corresponding to the conditional probability that the randomly generated probability falls into is selected as a new state.

The algorithm of Gibbs sampling based simulated annealing for minimising our energy function is presented in Algorithm 6. The iteration number I in the below algorithm is set empirically according to experience.

Algorithm 6 Gibbs Sampling Based Annealing Algorithm

(1.) Initialise a configuration x randomly, $\{x_i, i \in M\}$;
while $t = 1, \cdots, I$ where I is the iteration number **do**
(2.1.) Sequentially visit (or randomly choose) a node i, and carry out the sampling according to the conditional probability at node i as defined in Equation (6.2), and change its state from x_i to x_i' where x_i and x_i' are the current and new states at node i.
(2.2.) Let us define the error rate ϵ as the ratio of the incorrect matches to total matches. Check the error rate ϵ. Let L be the expected error rate according to experience. If $\epsilon < L$, exit. Otherwise, continue.
end while

6.1.2 Metropolis Sampler

The Metropolis sampler was developed by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller in 1953 [102], and it sometimes is called $M(RT)^2$ algorithm. The sampler takes a random walk to generate a set of samples that are distributed according to some arbitrary given function. The Metropolis sampler shares many of the characteristics of the Gibbs sampler, but is more generally applicable, as it avoids any need to sample from complicated distributions. It can be applied to problems where the state is either continuous or discrete, as long as it is possible to compute the ratio of the probabilities, or probability densities, of two states [111].

Let $H : \Omega \rightarrow R$ be an energy function where Ω is the configuration space of a Markov random field. We want to sample from the associated Gibbs distribution, and to minimise the energy function H by using the Metropolis sampler.

Let x be a configuration in the configuration space Ω. x is to be updated. In the Metropolis algorithm, a new configuration x' from x is created randomly. If x' is better than x in terms of H that is $H(x') \leq H(x)$ then it is accepted as the new configuration. If it is not, then it still gets a chance to be accepted. This enables the algorithm to escape local minima. A new configuration x' which is less favourable than x is not rejected automatically but accepted with a probability decreasing with the increment of energy $H(x') - H(x)$. This will allow the sampling algorithm to climb hills in the energy landscape and thus escape local minima [152]. If we only allow a new configuration whose energy is less than or equal to the current energy to be accepted, this is steepest descent. In this case, if the algorithm is trapped in a local minimum, it can not escape. If we allow a new configuration whose energy is greater than the

current energy to be accepted with a probability, then the algorithm has a chance to climb hills therefore has a chance to escape local minima. The Metropolis algorithm can be defined as follows [152].

(1) The proposal step: Given x, x' is sampled randomly based on x and is proposed as the new element.

(2) The acceptance step:

(a) If $H(x') \leq H(x)$ then x' is accepted as the new configuration.

(b) If $H(x') > H(x)$ then x' is accepted with probability $exp(H(x) - H(x'))$.

6.1.2.1 Annealing with Metropolis Sampler

The Metropolis sampler based simulated annealing allows a probabilistic acceptance of a randomly generated new configuration based on the energy difference [102]. Let $H(x)$ and $H(x')$ be the energy corresponding to the current state and new state respectively. If $H(x) > H(x')$, then the new state is accepted and becomes the current state. Otherwise, we associate a probability with the new state according to the energy difference as $exp(-\frac{H(x')-H(x)}{T_n})$. The new state associated with the energy close to the current state has high probability of being accepted; otherwise, the probability will be lower.

The acceptance probabilities for the Metropolis based simulated annealing are

$$P(x_i'|N(i)) = \begin{cases} 1 & if H(x') \leq H(x); \\ e^{-\frac{1}{T_n}(H(x')-H(x))} & if H(x') > H(x), \end{cases} \tag{6.3}$$

where x_i' is the new state at node i.

The algorithm for the Metropolis sampler based annealing is shown in Algorithm 7. The iteration number I in the below algorithm is set empirically according to experience.

In addition, corresponding to the sampling based simulated annealing paradigm, we devise a Las Vegas type algorithm, which has low order polynomial time complexity. The algorithm is described in the next section.

Algorithm 7 Metropolis Sampling Based Annealing Algorithm

(1.) Initialise a configuration x randomly, $\{x_i, i \in M\}$;
while $t = 1, \cdots, I$ where I is the iteration number **do**
(2.1.) Sequentially visit (or randomly choose) a node i, and randomly change its state from x_i to x_i'. Conduct acceptance of the new state x_i' according to the acceptance probabilities defined in Equation (6.3).
(2.2.) Let us define the error rate ϵ as the ratio of the incorrect matches to total matches. Check the error rate ϵ. Let L be the expected error rate according to experience. If $\epsilon < L$, exit. Otherwise, continue.
end while

6.2 Las Vegas Optimisation Algorithm

We define the conditional potential for a single node i as

$$P(x_i|N(i)) = exp(-[h_1(x_i) + w * \sum_{j \in N(i)} h_2(x_i, x_j)]), \tag{6.4}$$

where, h_1 and h_2 are defined as in Section 6.1.1.1. w is the balancing factor between the similarity measure of Gaussian curvature and the pairwise distance defined on the 2-element cliques.

The main procedure of the Las Vegas optimisation algorithm is to calculate conditional potentials for the current state and all other possible states at a node i. According to the calculated conditional potentials, put all the states whose conditional potentials are greater than the current state's conditional potential into a candidate pool. After this, randomly select one state from the candidate pool to be the new state for node i. Check whether the algorithm converges to a local minimum. If yes, terminate the iteration and restart with another randomly initialised configuration x. We check whether the algorithm converges to a local minimum according to follows. If after continuous five iterations the algorithm has no progress and has not reached the expected error rate, then the algorithm is considered as having converged to a local minimum.

The Las Vegas algorithm [110] for the optimisation problem is shown in Algorithm 8. The iteration number I is defined as before.

In our experiments, the optimisation algorithm gives the correct results in polynomial time

Algorithm 8 Las Vegas Algorithm [110]

(1.) Initialise a configuration x randomly, $\{x_i, i \in M\}$;
while $t = 1, \cdots, I$ where I is the iteration number **do**
  while Traversing the field M, for each individual node i **do**
    (2.1.1.) Calculate the conditional potential for the current state of i $P(x_i|N(i))$ according to Equation (6.4);
    (2.1.2.) Use the same formula to calculate the conditional potentials $P(X_i|N(i))$ for all the potential states of i over the state space;
    (2.1.3.) Put all the states whose conditional potentials $P(X_i|N(i)) > P(x_i|N(i))$ into a candidate pool;
    (2.1.4.) Randomly select one state from the candidate pool to be the new state for the current node i;
  end while
  (2.2.) Let us define the error rate as the ratio of incorrect matches to total matches. Check whether the error rate is acceptable. If yes, terminate the iteration; otherwise, continue;
  (2.3.) Check whether the algorithm converges to a local minimum. If yes, terminate the iteration and go back to 1 to restart with another randomly initialised configuration x; otherwise, continue.
end while

in all cases.

The time complexity of the algorithm is $O(kmn)$, where k is the number of iterations, m is the size of the Markov random field, and n is the size of the state space. In certain number of iteration k, the program traverses the Markov random field, and also traverses the whole state space to calculate the conditional potentials in each iteration. Therefore, the complexity is $O(kmn)$.

6.3 Evaluation

In this section, we give introduction on data set and evaluation of the Las Vegas optimisation algorithm. We also present the comparison of simulated annealing and Las Vegas and the comparison of Gibbs sampler and Metropolis sampler based simulated annealings.

6.3.1 Data Set

The experimental data we use is a triangular surface mesh as shown in Figures 6.1 and 6.2. The triangular surface mesh is from [43] with 448 faces and 226 vertices, which is publicly

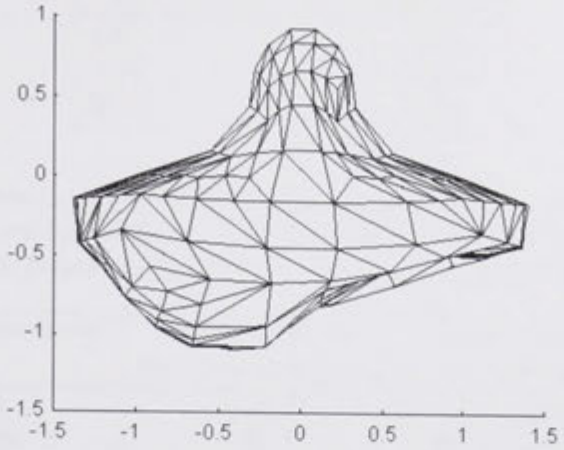


Figure 6.1: Triangular mesh (mesh model).

Figure 6.2: Triangular mesh (rendering).

available at `http://www.cse.wustl.edu/~cmg/meshes.php`. Gaussian curvatures are obtained by the estimated principal curvatures. The curvature is disturbed with added noise that is random and uniformly distributed, with distribution intervals of [-5%,5%], [-10%,10%], and [-50%,50%] of the Gaussian curvature at each node respectively. All our experiments are carried out on a computer with Intel Pentium(R) 4 CPU 3.20GHz.

6.3.2 Las Vegas Optimisation Algorithm

We performed 100 experiments for each noise level of 5%, 10%, and 50%. The matching results are summarised in Table 6.1. Each time we start an experiment with a random initialisation of the configuration. We obtain the correct rate of node correspondence based on the

Curvature Disturbance	Error Rate
5%	0.0%
10%	8.0%
50%	28.0%

Table 6.1: Correspondence matching results by using Las Vegas algorithm with curvature disturbance levels up to 5%, 10%, and 50%. Error rate is defined as the ratio of the number of wrong point correspondences based on the ground truth to the total number of point correspondences.

Noise Level	Iteration Number for Convergence Loops	Range of Time to Converge
5%	30–40	20–90 seconds
10%	30–50	20–140 seconds
50%	25–40	4–30 minutes

Table 6.2: Running time and iteration numbers while performing matching using Las Vegas algorithm.

known ground truth. We define the correct rate as the ratio of correct matches to total matches. The Las Vegas algorithm always reaches the expected matching rates within some range of time.

The running times and iteration numbers are shown in Table 6.2. For curvature noise levels up to 5% and 10%, the running time is normally around one minute; for noise level of 50%, it takes several minutes to half an hour to converge. Let us define a convergence loop to be the loop in which the program reaches the expected matching rate. The expected matching rate is the matching rate we expect to reach and is set according to experience. A convergence loop is the outer loop in the Las Vegas algorithm associated with randomly initialising a configuration x. The outer loop in the Las Vegas algorithm checks whether the algorithm converges to a local minimum. The iteration number for a typical convergence loop is from 25 to 50. Some of the matching results for the curvature disturbance level of 50% are shown in Figure 6.3. From the figure, we can see that upper part matches to upper part.

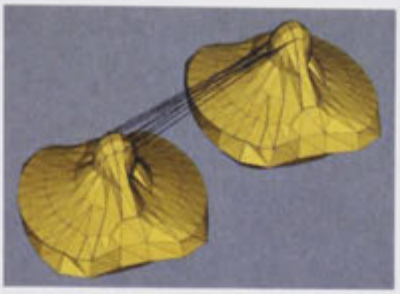


Figure 6.3: Matching between two surfaces at the noise level of 50%. Upper part matches to upper part.

Noise Level	Expected Matching Rate
5%	0.0%
10%	8.0%
50%	28.0%

Table 6.3: The expected matching rates for noise levels 5%, 10%, and 50%. The expected matching rate is the error rate we expect for the matching experiments, which is set empirically according to experience.

6.3.3 Simulated Annealing

6.3.3.1 Comparison of Simulated Annealing and Las Vegas

We compare simulated annealing algorithms with the derived corresponding Las Vegas type randomised algorithm while performing the 3D shape matching as follows. The experimental results show that the Las Vegas algorithm has performance comparable with the Gibbs sampler based and Metropolis sampler based simulated annealing algorithms, and sometimes it is faster than the Gibbs sampler based algorithm at the disturbance levels of 5% and 10%. It always gives correct results. Let us define the expected matching rate as the error rate we expect for the matching experiments, which is set empirically according to experience. The expected error rates for noise levels 5%, 10%, and 50% are shown in Table 6.3. Both Gibbs sampler and Metropolis sampler based annealing algorithms sometimes can not reach the expected matching rates within constrained time requirement.

6.3.3.2 Comparison of Gibbs Sampler and Metropolis Sampler Based Simulated Annealings

For simulated annealing, we experimented with both sequential visiting and random visiting schemes. For Gibbs sampler and Metropolis sampler based simulated annealings, as the sweeps and random visits increase, the results will become finer. However, we can not let the program run forever. Therefore, we need to set the resource limit restriction to restrict the number of sweeps and random visits while running the simulated annealing. The resource limit restriction we set for the Gibbs sampler based simulated annealing is 1,000 sweeps for sequential visiting scheme, and 200,000 random visits for random visiting scheme. Similarly, for the Metropolis based simulated annealing, 20,000 sweeps for sequential, and 4,000,000 visits for random scheme. We do not set the resource limit restriction to be the same because if we set the resource limit restriction lower as for Gibbs sampler, then all the experiments for Metropolis sampler can not reach the expected error rates; if we set the resource limit restriction higher as for Metropolis sampler, then the running time for Gibbs sampler would be too long. Therefore, we set the resource limit restriction for both Gibbs sampler and Metropolis sampler based simulated annealings according to the resource estimate to reach the expected matching rates. The expected matching rates are defined in Section 6.3.3.1.

We have performed 20 matching experiments for each noise level for both Gibbs sampler and Metropolis sampler based simulated annealings under both the sequential and random schemes. The comparison results between Gibbs sampler and Metropolis sampler based simulated annealing with sequential and random schemes under the noise levels of 5%, 10%, and 50% are shown in Tables 6.4, 6.5, and 6.6 respectively. The Gibbs sampler based simulated annealing is faster than the Metropolis sampler based one.

According to the experiments, some experiments can reach the expected error rates 0.0%, 8.0%, 28.0% for noise levels of 5%, 10%, and 50% respectively, and some can not. The rate of getting expectation, that is, the rate the experiments reach the expected matching rates, is summarised in the above tables.

From the experiments, for the Gibbs sampler based simulated annealing, the sequential

Sampler	Gibbs Sampler		Metropolis Sampler	
Visiting Scheme	Sequential	Random	Sequential	Random
Rate of Getting Expectation	95%	80%	70%	60%
Avg Sweep No	188	37825	9152	2003856
Avg Time (sec)	38	55	101	371

Table 6.4: Comparison of Gibbs sampler and Metropolis based simulated annealings at the noise level of 5%.

Sampler	Gibbs Sampler		Metropolis Sampler	
Visiting Scheme	Sequential	Random	Sequential	Random
Rate of Getting Expectation	90%	85%	50%	90%
Avg Sweep No	187	35928	6778	1062795
Avg Time (sec)	32	63	55	209

Table 6.5: Comparison of Gibbs sampler and Metropolis based simulated annealings at the noise level of 10%.

visiting scheme is better than the random visit scheme in terms of both the rate of getting expectation and the average running time. For the Metropolis sampler based simulated annealing, the random visiting scheme is better in terms of the rate of getting expectation in the curvature disturbance levels of 10% and 50%, and roughly has the rate in the level of 5%. However, its average running time is longer than the sequential visiting scheme.

For the error cases of the simulated annealing that can not reach the expected matching rate, this may be incomplete convergence which could be corrected by relaxing the resource limit. However, the algorithm can take 6 to 7 hours to reach the expected matching rate.

Sampler	Gibbs Sampler		Metropolis Sampler	
Visiting Scheme	Sequential	Random	Sequential	Random
Rate of Getting Expectation	95%	80%	20%	40%
Avg Sweep No	193	52656	15192	1814944
Avg Time (sec)	35	95	157	334

Table 6.6: Comparison of Gibbs sampler and Metropolis based simulated annealings at the noise level of 50%.

6.4 Summary

In this chapter, we have explored the use of randomised algorithms to minimise the energy function for 3D shape matching. We have investigated the Gibbs sampler and Metropolis sampler based simulated annealing algorithms, and also a Las Vegas type algorithm to minimise the energy function. From the experiments, the randomised algorithms are effective ways to minimise the energy function for 3D shape matching which has the number of states up to 200. Compared to belief propagation applied to the same synthetic data, the running time is similar while achieving the same error rate. However, the randomised algorithms have difficulty while being applied to minimise the energy function of 3D shape matching which has larger number of states, for example, 2000. Since all of our real hippocampus data sets have more than 2000 nodes, we can not apply the randomised algorithms to the shape matching for the hippocampus. We have addressed this large state space minimisation problem by using belief propagation (BP) and applied the BP algorithms to the shape matching for the hippocampus in Chapter 5.

Chapter 7

3D Rigid Body Registration

In this chapter, we present 3D rigid body registration techniques based on shape matching. 3D point correspondence matches are obtained from 3D shape matching using the belief propagation technique as described in Chapter 5. Using the 3D point correspondence matching results, a three point based transformation estimation approach is combined with the RANdom SAmple Consensus (RANSAC) to remove outliers, and obtain an inlier support set. Optimal transformation estimation methods based on eigenvalue decomposition, singular value decomposition, and quaternions are also given to obtain the final registration results.

Although 3D rigid body registration is an important supporting component to the thesis, it is not a key focus. However, the contents of this chapter are described for completeness of the description of the approach we used in the thesis. There are two related problems for the estimation of 3D rigid body transformation. The first problem is the transformation estimation techniques based on three pairs of corresponding points. We introduced two methods, one the coordinate system construction based approach and the other the critical point based approach, which are both described in Section 3.4 in Chapter 3. The second problem is the optimal transformation estimation method based on inliers, which will be presented in Section 7.4 of this chapter.

In all of the following, whenever we talk about point correspondences, or corresponding or matching points, we mean the point matchings obtained from performing BP under an MRF framework as described in Chapter 4 and 5.

7.1 3D Rigid Body Transformation Estimation

3D shape registration has wide applications in range image, statistical shape analysis, and medical imaging. Currently, the Iterative Closest Point (ICP) algorithm is one of the most popular approaches for 3D rigid body registration. However, ICP needs to have a good initialisation in order to obtain the global optimal solution, otherwise, it could converge to a local minimum [19].

In this chapter, we will present a global registration approach which does not require any initialisation. The 3D rigid body registration method is purely dependent on the point-wise correspondence between two meshed surfaces obtained by 3D shape matching. The 3D matching results are obtained by performing belief propagation under a Markov random field (MRF) framework as described in Chapter 5.

From the 3D shape matching results, we use a three-point based 3D rigid body transformation estimation technique and combine it with a RANSAC[41]-like algorithm to eliminate the correspondence outliers and get the inlier support set. After obtaining the inlier support set, we obtain an optimal 3D rigid body transformation estimation using techniques based on eigenvalue decomposition, singular value decomposition, and quaternions. This approach is standard in computer vision but less often used in medical image analysis.

7.2 3D Rigid Body Transformation Estimation from Three Pairs of Corresponding Points

Given two data sets M and M', which we seek to register, we first find the point correspondences using BP as in Chapter 5. From these point correspondences, we perform 3D rigid body transformation estimation using the coordinate system construction based three-points algorithm as presented in Section 3.4 of Chapter 3.

7.3 Inlier Support Seeking Algorithm

RANSAC stands for RANdom SAmple Consensus. It is a general parameter estimation approach designed to cope with a large proportion of outliers in the input data. It iteratively estimates parameters of a model from the observed data containing outliers [41]. RANSAC is a resampling technique that generates candidate solutions by using the minimum number of data points required to estimate the underlying model parameters.

We use a RANSAC-like algorithm to eliminate correspondence outliers and to obtain the correspondence inlier support. The minimum number of data points to instantiate the free parameters for estimating the 3D rigid body transformation of rotation and translation in our case is three.

Given two data sets $M = \{v_{l,i}\}$ and $M' = \{v_{r,i}\}$, the point correspondences between M and M' are given by $\{map(v_{l,i})\}$. To perform the RANSAC procedure, we randomly select three non-collinear point correspondences. From the three non-collinear point correspondences, $v_{l,1}, v_{l,2}, v_{l,3} \in M$, with their corresponding matches found by BP, $v_{r,1}, v_{r,2}, v_{r,3} \in M'$, we estimate the rotation R and translation t using the techniques described in Section 3.4. Then we apply the estimated rotation and translation to the data set M and transform it to the data set $\hat{M} = \{T(v_{l,i})\}$, and calculate the Euclidean distances between all the point pairs $\{T(v_{l,i}), map(v_{l,i})\}$. The inliers and outliers are finally determined from a predefined threshold.

The algorithm to locate inlier support is described in Algorithm 9.

Other approaches to inlier selection are also possible, for example, the recent related work on inlier selection and registration proposed by Albarelli et al [5, 4, 3] and Rodola et al [122] by using game theory followed by Torsello et al's work on applying game theory to grouping [139].

Algorithm 9 Inlier Support Seeker

1. Let two data sets $M = \{v_{l,i}\}$ and $M' = \{v_{r,i} = map(v_{l,i})\}$, where $i = 1, ..., m$, $m = |M|$ and $map(\cdot)$ denotes the matching function obtained by BP.
2. Let I_n be the calculated inlier support number, and ω be the expected inlier support number.

while $I_n \leq \omega$ **do**

3.1 Randomly select three non-collinear points $v_{l,1}, v_{l,2}, v_{l,3} \in M$ whose corresponding points in M' are $v_{r,1}, v_{r,2}, v_{r,3}$ respectively.

3.2 Estimate rotation matrix R and translation vector t using method in Section 3.4 to give the transformation T(R,t).

3.3 Apply the transformation $T(R, t)$ to the points in M giving

$$\hat{M} = \{T(v_{l,i})\}, i = 1, ..., m.$$

3.4 Calculate the Euclidean distances between the pairs $(T(v_{l,i}), v_{r,i})$

$$d_i = |T(v_{l,i}), map(v_{l,i})|, i = 1, ..., m.$$

3.5 Check whether the Euclidean distance for a correspondence is less than a predefined threshold τ.

If

$$d_i < \tau,$$

where, $i = 1, ..., m$, the correspondence is considered as an inlier;

otherwise, an outlier.

end while

7.4 Optimal Estimation of Transformation from Inlier Support

After obtaining the inlier support data set, the final 3D rigid body transformation of rotation, translation, and scaling factor can be estimated based on the correspondences of the inlier support data set.

We investigated and implemented three methods to estimate the transformation, namely, (1) eigenvalue decomposition; (2) singular value decomposition; and (3) quaternions. When we performed optimal estimate of the transformation, in order to compare the three methods effectively, we used the same inlier support set (the average is the same) for all the three methods. From the estimation experiments, the three methods give almost the same results with the difference starting from the fifth digit after the decimal point for estimation of the rotation matrix R.

The transformation was estimated by the following steps.

1. We used the coordinate system construction based three-points algorithm as described in Section 3.4 to obtain the initial 3D rigid body transformation estimation, the rotation R and translation t, from three pairs of corresponding points;

2. Combined the RANSAC technique with the result from step 1 to eliminate outliers and obtain the inlier support set as described in Section 7.3;

3. The inlier support set was used for the input for the final optimal estimation of the transformation.

7.4.1 Eigenvalue Decomposition

Suppose the inlier support sets we have found are $\mathcal{I}_l = \{v_{l,i}\}$, $v_{l,i} \in M$, with $|\mathcal{I}_l| = n$, and $\mathcal{I}_r = \{v_{r,i} = map(v_{l,i})\}$, where $i = \{1, ..., n\}$ and $map(\cdot)$ denotes the matching function obtained by BP. The centroids corresponding to M and M' are obtained as

$$\bar{c}_l = \frac{1}{n}\sum_{i=1}^{n} v_{l,i},$$

$$\bar{c}_r = \frac{1}{n}\sum_{i=1}^{n} v_{r,i}.$$

Then the new coordinates for the inlier support sets relative to the centroids will be

$$\begin{aligned} v'_{l,i} &= v_{l,i} - \bar{c}_l, \\ v'_{r,i} &= v_{r,i} - \bar{c}_r, \end{aligned}$$

where $i = \{1, ..., n\}$.

Let

$$\begin{aligned} h_{xx} &= \sum_{i=1}^{n} x'_{r,i} x'_{l,i}, \\ h_{xy} &= \sum_{i=1}^{n} x'_{r,i} y'_{l,i}, \\ &\dots, \end{aligned}$$

where $x'_{l,i}$ and $x'_{r,i}$ are the x coordinates of points $v'_{l,i}$ and $v'_{r,i}$ respectively.

Then we have

$$H = \begin{bmatrix} h_{xx} & h_{xy} & h_{xz} \\ h_{yx} & h_{yy} & h_{yz} \\ h_{zx} & h_{zy} & h_{zz} \end{bmatrix}. \tag{7.1}$$

Therefore we can estimate the rotation matrix as in [61]

$$R = H(H^T H)^{-1/2}. \tag{7.2}$$

The corresponding translation from M to M' can be estimated as

$$t = \bar{c}_r - sR\bar{c}_l. \tag{7.3}$$

The scaling factor s can be calculated by

$$s = \sum_{i=1}^{n} v'_{r,i} \cdot Rv'_{l,i} \Big/ \sum_{i=1}^{n} \|v'_{l,i}\|^2. \tag{7.4}$$

From the estimated rotation, translation, and scaling factor, we register the points in M to

the points in M' by performing

$$sRv_{l,i} + t.$$

For the actual parameter calculation with eigenvalue decomposition approach, we first calculate h_{xx}, h_{xy}, ..., from the coordinates of all the inliers. Then we construct H matrix according to Equation (7.1). The rotation matrix is obtained from Equation (7.2), and the translation is calculated according to Equation (7.3).

7.4.2 Singular Value Decomposition

The optimal estimation of R and t can be obtained based on singular value decomposition (SVD) according to Arun, Huang, and Blostein [8].

The H matrix definition is the same as Equation (7.1). We calculate the SVD of H

$$H = UWV^t, \tag{7.5}$$

where $U, V, W \in \mathbb{R}^{3\times3}$. U and V are orthonormal, and W is diagonal with nonnegative elements.

Let

$$Y = VU^t. \tag{7.6}$$

Calculate the determinant of Y, $det(Y)$, and check the following conditions,

- if $det(Y) = +1$, then the estimated rotation is $R = Y$; and
- if $det(Y) = -1$, the algorithm fails.

In the case when $det(Y) = -1$, Y is a reflection. This degenerate case usually does not occur [8]. If it does occur, we can divide the situation into two cases, noiseless and noisy. In noiseless case, if Y is a reflection, then Y^t is a rotation, and vice versa. Therefore, the rotation matrix can be estimated by $Y^t = V^tU^t$. In noisy case, this can only happen when the noise is very large. In that case, the lease square solution is probably useless [8].

The translation t and scaling factor s can be estimated similarly with Equation (7.3) and (7.4) in Section 7.4.1.

For the actual parameter calculation with singular value decomposition approach, we first perform SVD of the same matrix H as in the eigenvalue decomposition approach of Section 7.4.1 according to Equation (7.5). Then we calculate matrix Y according to Equation (7.6). If the determinant of Y is 1, then the rotation matrix R equals to Y. The calculation of the translation t is the same as in Section 7.4.1 for eigenvalue decomposition approach.

7.4.3 Quaternions

A quaternion can be thought of as a vector with four components, as a composite of a scalar and an ordinary vector, or as a complex number with three different imaginary parts. Quaternions will be denoted here by using symbols with a dot above them. By using complex number notation, we have

$$\dot{q} = q_0 + iq_x + jq_y + kq_z,$$

a quaternion with real part q_0 and three imaginary parts, q_x, q_y, and q_z.

The quaternion has the following properties

$$i^2 = -1, j^2 = -1, k^2 = -1;$$

$$ij = k, jk = i, ki = j;$$

$$ji = -k, kj = -i, ik = -j.$$

The dot product of two quaternions is the sum of products of corresponding components

$$\dot{p} \cdot \dot{q} = p_0 q_0 + p_x q_x + p_y q_y + p_z q_z.$$

The square of the magnitude of a quaternion is the dot product of the quaternion with itself

$$\|\dot{q}\|^2 = \dot{q} \cdot \dot{q}.$$

A unit quaternion is a quaternion whose magnitude equals 1. The conjugate of a quaternion is obtained by negating its imaginary part

$$\dot{q}^* = q_0 - iq_x - jq_y - kq_z.$$

As shown by Horn [60], the rotation of 3D rigid body transformation can be represented as a unit quaternion. Let $\dot{r} = [r_0 r_x r_y r_z]^T$ be a unit quaternion which represents the rotation, where $r_0 \geq 0$, and $r_0^2 + r_x^2 + r_y^2 + r_z^2 = 1$. The unit rotation quaternion can be obtained by maximising the following quadratic form

$$\dot{r}^T Q \dot{r}, \tag{7.7}$$

subject to $\dot{r} \cdot \dot{r} = 1$, where $Q \in \mathbb{R}^{4\times4}$ and is defined below.

Let h_{xx}, h_{xy}, and h_{xz}, $\cdots$ be the same elements as defined in Equation (7.1) for eigenvalue decomposition based approach. We construct a symmetric matrix Q as follows,

$$Q = \begin{bmatrix} h_{xx} + h_{yy} + h_{zz} & h_{yz} - h_{zy} & h_{zx} - h_{xz} & h_{xy} - h_{yx} \\ h_{yz} - h_{zy} & h_{xx} - h_{yy} - h_{zz} & h_{xy} + h_{yx} & h_{zx} + h_{xz} \\ h_{zx} - h_{xz} & h_{xy} + h_{yx} & -h_{xx} + h_{yy} - h_{zz} & h_{yz} + h_{zy} \\ h_{xy} - h_{yx} & h_{zx} + h_{xz} & h_{yz} + h_{zy} & -h_{xx} - h_{yy} + h_{zz} \end{bmatrix}. \tag{7.8}$$

The unit quaternion that maximises the Equation (7.7) is the eigenvector of Q associated with the most positive eigenvalue, which can be obtained by

$$det(Q - \lambda I) = 0,$$

where $I \in \mathbb{R}^{4\times4}$ is the identity matrix.

After obtaining the largest positive eigenvalue λ_m, the corresponding eigenvector $\dot{r}_m$ is

calculated by solving the homogeneous equation

$$[Q - \lambda_m I]\dot{r}_m = 0.$$

From the estimated unit rotation quaternion $\dot{r}$, for a purely imaginary quaternion $\dot{v}_{l,i}$, which denotes a point in three dimensional space in the left data set, the rotation of the point to the right is obtained by

$$\hat{\dot{v}}_{r,i} = \dot{r}\dot{v}_{l,i}\dot{r}^*,$$

where $\hat{\dot{v}}_{r,i}$ is a purely imaginary quaternion also, which represents a point in three dimensional space in the right coordinate system.

The estimated unit rotation quaternion $\dot{r}$ can also be transformed to an orthonormal rotation matrix $R \in \mathbb{R}^{3\times3}$ as follows,

$$R = \begin{bmatrix} (r_0^2 + r_x^2 - r_y^2 - r_z^2) & 2(r_x r_y - r_0 r_z) & 2(r_x r_z + r_0 r_y) \\ 2(r_y r_x + r_0 r_z) & (r_0^2 - r_x^2 + r_y^2 - r_z^2) & 2(r_y r_z - r_0 r_x) \\ 2(r_z r_x - r_0 r_y) & 2(r_z r_y + r_0 r_x) & (r_0^2 - r_x^2 - r_y^2 + r_z^2) \end{bmatrix}. \tag{7.9}$$

The translation t and scaling factor s can be estimated similarly as in the eigenvalue decomposition based approach.

For the actual parameter calculation with quaternions, we first construct Q matrix from h_{xx}, h_{xy}, ..., as calculated in Section 7.4.1 for eigenvalue decomposition approach according to Equation (7.8). Then we calculate the eigenvector of Q corresponding to Q's most positive eigenvalue. The rotation matrix R is calculated from the obtained eigenvector according to Equation (7.9). The calculation of the translation t is the same as in Section 7.4.1 for eigenvalue decomposition approach.

7.5 Summary

From this chapter, a three point based transformation estimation technique is combined with RANSAC to eliminate the correspondence outliers and to obtain the inlier support set. The

point correspondences used to estimate the transformation are obtained from the 3D shape matching using belief propagation as described in Chapter 5. The input of 3D rigid body registration is obtained by performing 3D shape matching using belief propagation.

The three optimal estimation methods based on eigenvalue decomposition, singular value decomposition, and quaternions are used to obtain the final registration results. The results obtained by the three methods are the same. In the experiments, we use eigenvalue decomposition method to produce the 3D registration results in Chapter 8.

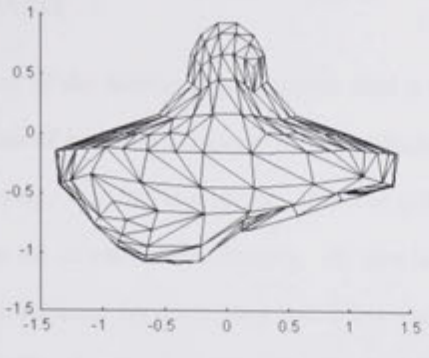
Chapter 8

Experimental Results

In this chapter, we present experimental results showing the recovery of dense correspondences with sparse update belief propagation with the integration of clamping. We also demonstrate 3D rigid body registration results by using 3D shape matching outcomes. We have used both synthetic data and real data to perform the experiments for matching with sparse update belief propagation and registration. In order to investigate the impact of alternative methods and parameter tuning on 3D shape matching, we perform a series of experiments: comparing max product with sum product algorithm by shape matching experiments; investigating the impact of scale of Gaussian curvature on matching; comparing results between sparse update and full update belief propagations; and investigating the impact of relaxation of neighbourhood constraints on matching.

Using synthetic data, we demonstrate the results of 3D rotation and translation estimation experiments of using the global rigid body registration method based on the shape matching. We then demonstrate 3D rigid body registration results using real human hippocampus data. We examine inlier support and the distribution of inliers over the whole space. We give comparison results of our method with the first order ellipsoid (FOE) based registration method. In addition, we show the impact on registration of using the clamping techniques. Finally, we present the topology consistency check results of 3D shape matching.

The registration results after clamping show almost all nodes as inliers. This is a high degree of topological consistency. Since the hippocampi have a large degree of shape variation across individuals, this shows high quality dense correspondences across quite deformed surfaces.



(a) Mesh model.

(b) Rendering.

Figure 8.1: Triangular mesh.

Compared to FOE, our method achieves comparable results and provides an alternative way to perform registration for 3D neuroanatomical objects. The advantage of our method is that the point-to-point correspondences obtained allow more local comparison and deformable matching between two 3D neuroanatomical objects.

8.1 Data Description and Computing Facility

8.1.1 Synthetic Data

The synthetic data we use to perform the 3D shape matching and registration experiments is a triangular surface mesh, which is publicly available at `http://www.cse.wustl.edu/~cmg/meshes.php`. The surface mesh has 448 faces and 226 vertices, and is shown in Figure 8.1. Principal curvatures at each node are available to use.

8.1.2 Real Data

In order to show that our 3D shape matching and registration techniques work across a broad range of population, we have performed 3D shape matching and registration experiments using human hippocampus data from both the Open Access Series of Imaging Studies (OASIS) project and the Personality and Total Health (PATH) through life project.

8.1.2.1 OASIS

The first part of the human hippocampus data is from a publicly accessible project, the Open Access Series of Imaging Studies (OASIS), which can be accessed at `http://www.oasis-brains.org/`. OASIS is a project aimed at making MRI data sets of the human brain freely available to the scientific community. Its aim is to facilitate future discoveries in basic and clinical neuroscience by compiling and freely distributing MRI data sets [96]. OASIS is made available by Dr. Randy Buckner at the Howard Hughes Medical Institute (HHMI) at Harvard University, the Neuroinformatics Research Group (NRG) at Washington University School of Medicine, and the Biomedical Informatics Research Network (BIRN).

The hippocampus data sets we use for matching and registration experiments are subjects with no dementing diagnosis (normal) aged in their 60s. The mean age of the subjects is 64.88. We take all the 24 data sets of normal subjects aged in their 60s to perform the registration experiments.

8.1.2.2 PATH

The second part of human hippocampus data is from the Personality and Total Health (PATH) Through Life Project. The PATH is a 20 year longitudinal study of 7500 adults randomly selected from the electoral rolls of Canberra and Queanbeyan, Australia. The owner of the PATH data set is the Centre for Mental Health Research (CMHR), the Australian National University. The aim of PATH is to investigate the causes of common mental health problems. A large collection of health data is gathered for each participant, and a subsample of 478 subjects aged between 60 and 64 participated in a Magnetic Resonance Imaging (MRI) study.

The hippocampus data sets we use for matching and registration experiments are from normal subjects aged between 60 and 64. We randomly select 21 hippocampi from 478 data sets in this age range to perform the registration experiments.

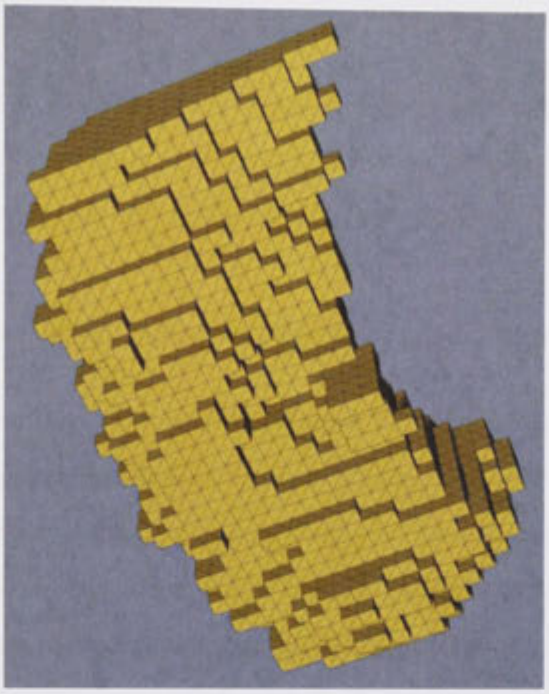


Figure 8.2: The quadrangular mesh representation of a hippocampus named 335 with 2544 nodes from PATH.

8.1.2.3 Handtracing and Quadrangular Mesh Representation

We use manually segmented hippocampus data sets by hand tracing from MRI scans of human brain for our experiments. The MRI scans were traced by an expert neurologist for the hippocampi according to the protocol outlined by [146]. Based on the manually segmented hippocampus 3D data sets, we construct the corresponding 3D surface meshes of the hippocampus. The hand segmented human hippocampi are represented as genus zero voxel-based quadrangular meshes according to the method proposed in [90]. The construction of the 3D surface meshes of the hippocampus is done by Paulette Lieby. The 3D shape matching and registration performed in this chapter are based on the 3D surface meshes of the hippocampus.

The quadrangular mesh representation of a hippocampus from PATH named 335 with 2544 nodes is shown in Figure 8.2.

8.1.3 Computing Facility

The computer we used for the 3D shape matching experiments with synthetic data and registration experiments is Intel Pentium(R) 4 CPU 3.20GHz. For the 3D shape matching experiments with real data, the computer we used is Intel Xeon(R) CPU 2.00GHz.

8.2 Comparison of Alternative Methods and Parameters

In this section, we investigate the impact of alternative methods and parameters on 3D shape matching. We investigate the relationship between different choices of methods and different settings of parameters and shape matching by using both synthetic data and real data. The comparisons performed are max versus sum product, size of neighbourhood, scale of curvature, sparse versus full update, and relaxation of neighbourhood constraints.

8.2.1 Max versus Sum Product

In order to compare the sum product algorithm with the max counterpart in 3D shape matching, we run the algorithms on the synthetic data as described in Section 8.1.1. Gaussian curvature is obtained from the estimated principal curvatures available in the data set. We added uniform distributed curvature noise at each node in levels of 5%, 10%, and 50%, and the corresponding noise distribution intervals are [-5%,5%], [-10%,10%], and [-50%,50%] respectively.

The comparison of the sum and max product convergence property in the three different noise levels is shown in Figure 8.3. Error rate is defined as the ratio of incorrect correspondences to total matches.

Each iteration takes 2 seconds for both sum and max products. For the noise levels of 5% and 10%, the max-product takes 50 iterations to converge. There is an oscillating stage for the noise level of 10% after monotonically decreasing in the first few iterations, and finally it becomes stabilised after 50 iterations. For the noise level of 50%, the max product takes around 25 iterations to become stabilised. Both the 10% and 50% show some minor changes after being stabilised.

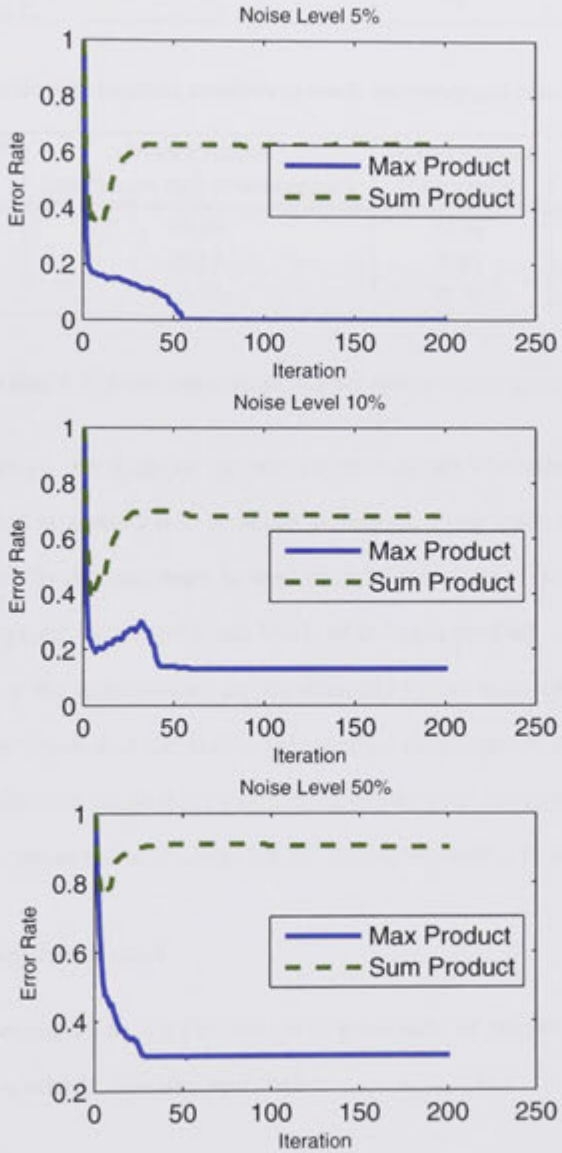


Figure 8.3: Convergence property of sum and max products.

Noise Level	Max-Product (minimum and convergence)	Sum-Product (minimum)	Sum-Product (convergence)
5%	50	10	25
10%	50	6	25
50%	25	4	25

Table 8.1: Iteration numbers to reach minimum and converge.

Noise Level	Max-Product (minimum and convergence)	Sum-Product (minimum)	Sum-Product (convergence)
5%	0.00	0.33	0.62
10%	0.13	0.41	0.68
50%	0.30	0.76	0.90

Table 8.2: Error rate comparison for sum and max products.

For the sum-product, the shape of the convergence curves is similar for all the three noise levels. All show that at around five to seven iterations, error rates reach their minimums, and then increase and finally converge around after 25 iterations. There are also some small changes after convergence for all the noise levels of the sum product.

The error rates for the sum-product are significantly higher than the max-product, both for the minimums it can reach and the stable values after convergence at all noise levels. The discussion on why the sum product approach does not achieve the results as good as the max product approach is conducted in Section 5.5.2.2. The comparison is shown in Table 8.2.

8.2.2 Size of Neighbourhood

We perform the experiments to test the impact of relaxation of neighbourhood constraints on matching with both synthetic data and real data.

8.2.2.1 Relaxation of Neighbourhood Constraints with Synthetic Data

We experimented with matching by relaxing the pairwise spatial assignment constraints, to tolerate a larger range of pairwise assignment on the meshed surface M'. We used synthetic data for experiments. We define layer 1 as $N(x_i) = \{x_j, x_j \text{ is the neighbour of } x_i\}$; layer 2 as $NN(x_i) = \{x_j, x_j \text{ is the neighbour of } N(x_i)\}$. In the experiments, we used a hierarchical

Assignment Range	Error Rate			Iteration Number		
Noise Level	5%	10%	50%	5%	10%	50%
Layer 1	0%	13%	30%	54	50	28
Layer 2	0%	8%	28%	70	45	10

Table 8.3: Error rate and iteration number comparison between layer 1 and layer 2 based on the pairwise potential function $\psi_{ij}(x_i, x_j)$ definition.

penalty function as introduced in Section 8.2.2.2 to measure the pairwise compatibility over the meshed surface M'. The corresponding hierarchical pairwise potential function $\psi_{ij}(x_i, x_j)$ over ij is defined as follows,

$$\psi_{ij}(x_i, x_j) = \begin{cases} 1, & \text{if } x_i \in N(x_j) \\ 0.1, & \text{if } x_i \in NN(x_j) \\ 0, & \text{otherwise,} \end{cases}$$

where $N(x_j)$ denotes neighbours of x_j, and $NN(x_j)$ denotes neighbours' neighbours of x_j.

The potential function gives a small penalty for pairwise nodes on an MRF to be assigned to layer 2 neighbours on its matched meshed surface M', a node with its immediate neighbours' neighbours. It favours layer 1 neighbours, a node with its immediate neighbours, with no penalty. For all other cases, it gives the highest penalty.

The comparison results of layer 2 with layer 1 based on the pairwise potential function $\psi_{ij}(x_i, x_j)$ definition is given in Table 8.3. Each iteration takes 15 seconds for layer 2 assignment. For the 50% noise level, it takes 10 iterations to reach the error rate under 28%; and the total time to converge is 150 seconds, longer than the layer 1, which takes 60 seconds to converge, but it can get slightly better matching results. For the 10% noise level, it takes similar number of iterations to layer 1, around 50 iterations, to get the error rate below 8%. For the 5% noise level, it takes around 70 iterations to get error free matching results. The experiments show that for higher noise levels, using layer 2 assignment could get slightly improved matching results without losing much efficiency.

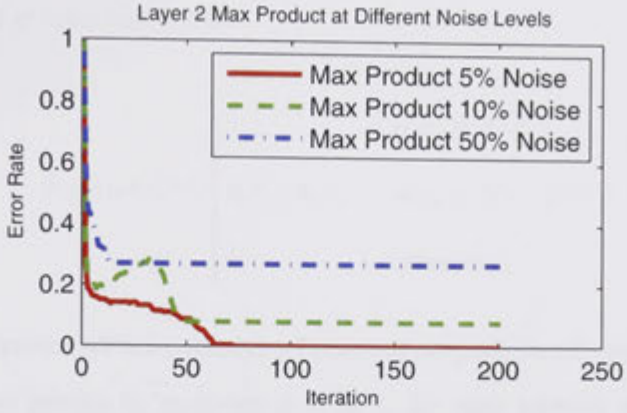


Figure 8.4: Layer 2 max product convergence property.

The convergence property for the layer 2 max product at different noise levels is shown in Figure 8.4.

8.2.2.2 Relaxation of Neighbourhood Constraints with Real Data

In the sparse update message update Equation (5.10), the pairwise potential function gives a strong constraint to the neighbourhood structure. In order to increase the matching result, we relax the neighbourhood constraints to allow different layers of neighbourhood structure to play a role in the message update equation. Accordingly, we further change the sparse update message update Equation (5.10),

$$m_{ij}(x_j) = \sum_{x_i \in N \cdots N(x_j)} \left[\phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i) \right], \tag{8.1}$$

where $N \cdots N$ is defined as arbitrary number of layers of neighbourhood. A layer is defined as the collection of boundary nodes' immediate neighbours as defined in Section 8.2.2.1. Typically in our application, we take 4 or 5 layers of neighbourhood.

We have experimentally relaxed the neighbourhood constraints up to neighbour layer 5 by using the real hippocampus data 0070 and 0135 from OASIS.

We assign the ψ value function as

$$\psi_{ij}(x_i, x_j) = \begin{cases} 1, & \text{if } x_i \in N(x_j) \\ 0.8, 0.6, \ldots, & \text{if } x_i \in N \cdots N(x_j) \\ 0, & \text{otherwise,} \end{cases} \tag{8.2}$$

where $N \cdots N$ denotes arbitrary number of layers of neighbourhood. For immediate neighbours, we give no penalty by assigning ψ to be 1; for other arbitrary number of layers of neighbours, we gradually increase the penalty by assigning ψ to be the values 0.8, 0.6, 0.4, and 0.2 for layer 2, 3, 4, and 5 respectively; for all other cases, we give the highest penalty by assigning ψ to be 0.

During the experiments, we obtained the inlier support set from the 3D rigid body registration. The matching versus neighbourhood and running time versus neighbourhood are shown in Figure 8.5 and Figure 8.6. After relaxation of neighbourhood constraints, we notice only a slight increase in the quality of the matching results, reflected by an increase of 2% in the inlier support set. This suggests that the restriction of belief over neighbourhoods only marginally impacts on the final converged state of the MRF. From the experiments, the inlier support set increases to neighbour layer 3, and then decreases to neighbour layer 5. The running time increases exponentially with the number of neighbourhood layers.

From the experiments of relaxation of neighbourhood constraints with both synthetic data and real data, we can see that relaxation of neighbourhood constraints only slightly increases the matching rate. The experiments with synthetic data and real data are consistent in verifying the slight increase of the matching rate.

8.2.3 Scale of Curvature

We have conducted experiments of 3D shape matching with real hippocampi of 0070 and 0135 from OASIS in different scales of Gaussian curvature. We ran 3D shape matching with eleven scales of Gaussian curvature including the original scale as shown in Figure 8.7. The experiments show that generally the inlier support set increases while the scale of Gaussian curvature

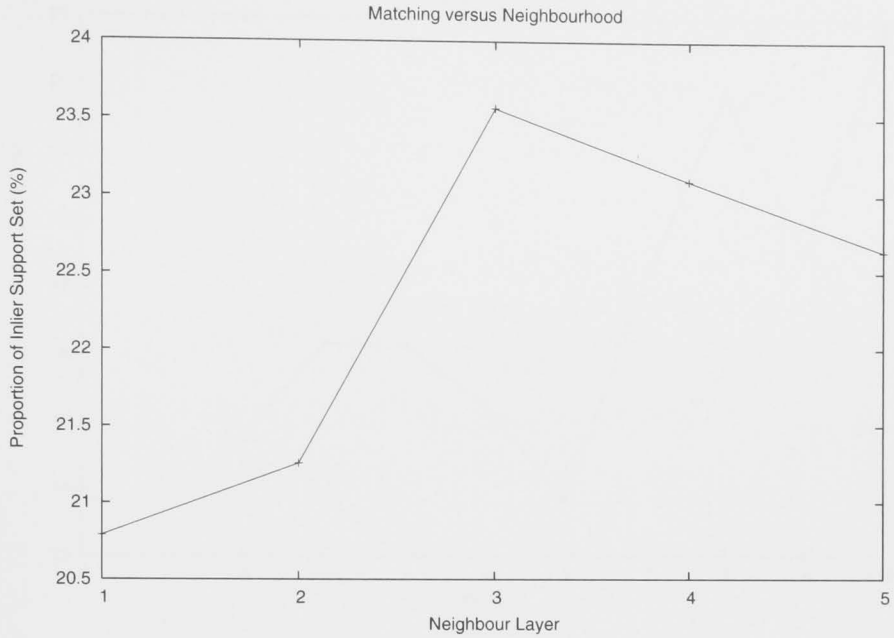


Figure 8.5: Inlier support set versus neighbourhood.

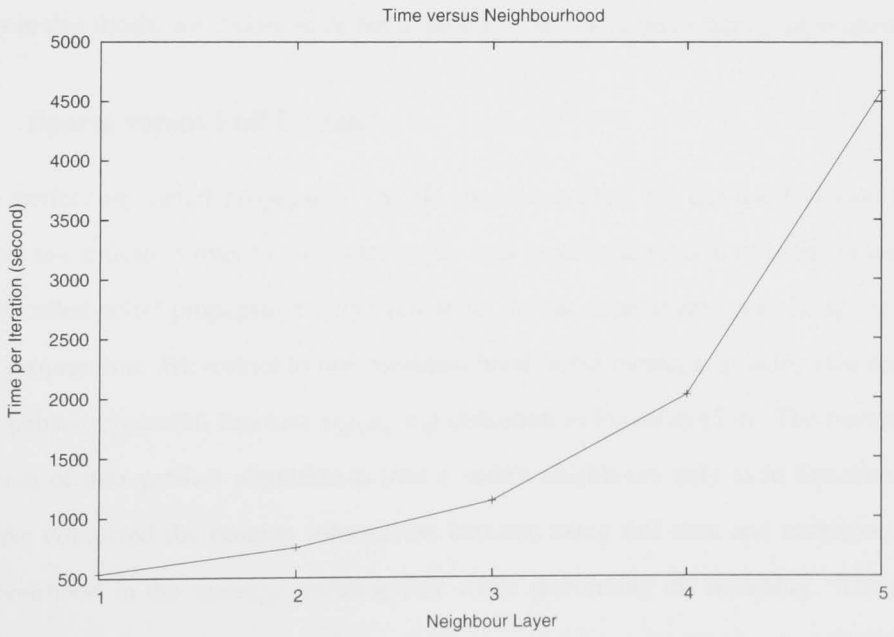


Figure 8.6: The running time versus neighbourhood.

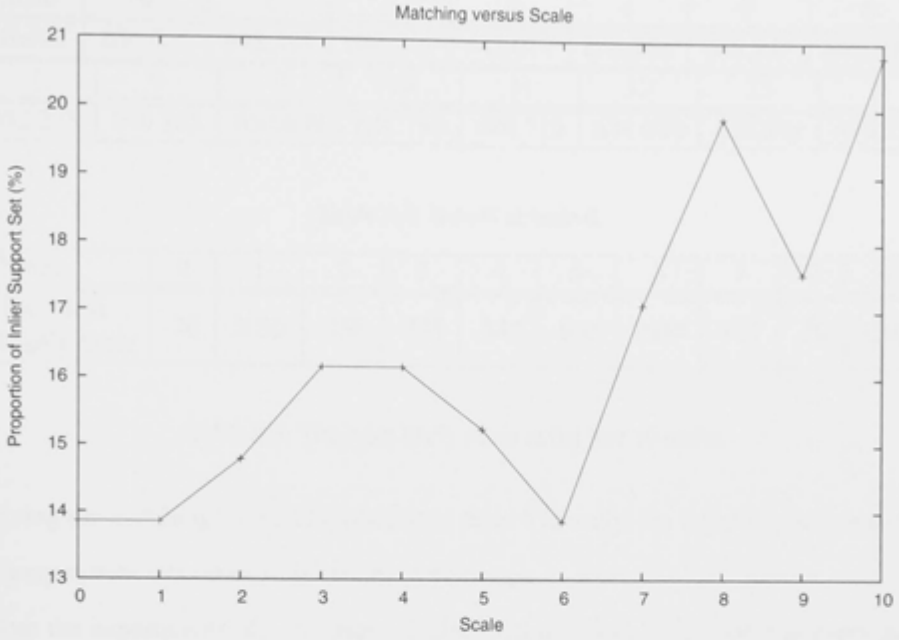


Figure 8.7: Matching versus the scale of Gaussian curvature.

becomes coarser. The inlier support set achieves the maximum in scale ten. For subsequent results in this thesis, we choose scale ten to perform the 3D shape matching experiments.

8.2.4 Sparse versus Full Update

While performing belief propagation for 3D shape matching, we can use Equation (5.8) to perform maximisation over the full state x_i for max-product algorithm of belief propagation. This is called belief propagation under full state. In our experiments, we use sparse update belief propagation. We restrict to one neighbourhood in the message updating rule according to the pairwise potential function $\psi_{ij}(x_i, x_j)$ definition in Equation (5.9). The maximisation operation of max-product algorithm is over a node's neighbours only as in Equation (5.11). We have compared the running information between using full state and restricting to one neighbourhood in the message updating rule while performing the matching. The running time for using full state is 16 hours for each iteration, and for restricting to one neighbourhood is 6 minutes per iteration. The sparse update BP is 160 times faster than the standard BP while

State	0	1	2	3	4	5	6
Belief	650.088	648.395	644.212	643.017	646.366	646.849	652.500
7	8	9	10	11	12	13	14
652.248	650.105	650.609	648.799	646.719	644.636	640.944	639.279

Table 8.4: Beliefs at node 0.

Node	0	1	2	3	4	5	6	7	8	9
The Most Likely State	76	2035	356	355	347	963	1034	667	38	1283

Table 8.5: The most likely states at the first 10 nodes.

performing the matching for real hippocampus data. Typically, the matching between two real hippocampus data sets takes 10 iterations to converge.

From the experiments, the messages, beliefs and the most likely states at every node for each iteration are the same. We have verified this by performing a comparison test. The comparison results are identical. The hippocampus data we use for the comparison test is 310 and 219 with 2618 and 2594 nodes respectively from the PATH. The first 15 beliefs corresponding to each state after one iteration at node 0 are shown in Table 8.4. The most likely states at the first 10 nodes after one iteration are shown in Table 8.5.

8.3 Matching and Clamping

In this section, we present the experiments of matching with sparse update BP and matching with clamping.

8.3.1 Matching with Sparse Update BP

By experiments, we have made choices of methods and parameters as summarised in Table 8.6. We will perform 3D shape matching according to the following choices of methods and parameters: max product belief propagation, scale 10 of curvature, sparse update message updating rule, and immediate neighbourhood constraint.

We have performed 3D correspondence matching experiments according to the above men-

Parameter	Max vs Sum	Scale of Curvature	Sparse vs Full Update	Neighbourhood
Choice	Max	Scale 10	Sparse	Immediate

Table 8.6: Summary of choice of methods and parameters.

tioned choices of methods and parameters using real human hippocampus data from both OASIS and PATH as described in Section 8.1.2. The matching result for real data of hippocampi with names of 0070 with 2164 nodes and 0135 with 2236 nodes from OASIS is shown in Figure 8.8. The matching result for real data of hippocampi with names of 118 with 2700 nodes and 124 with 2698 nodes from PATH is shown in Figure 8.9.

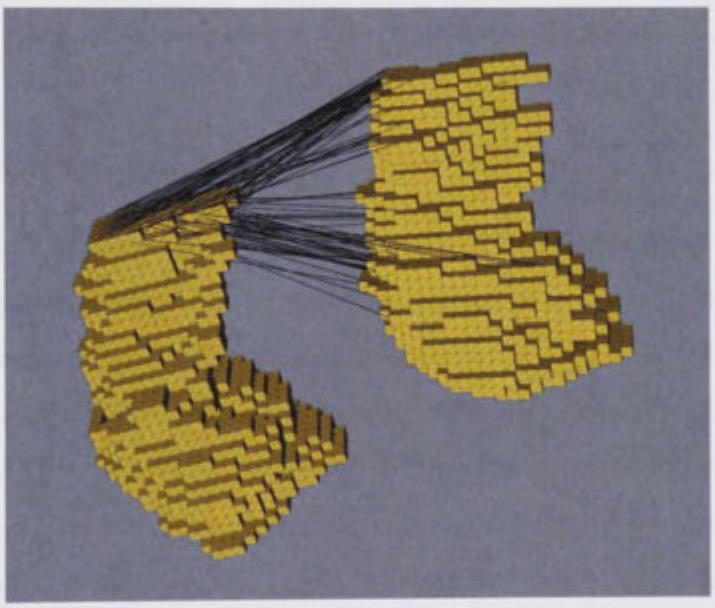


Figure 8.8: Matching of two real hippocampi 0070 and 0135 from OASIS with no clamping.

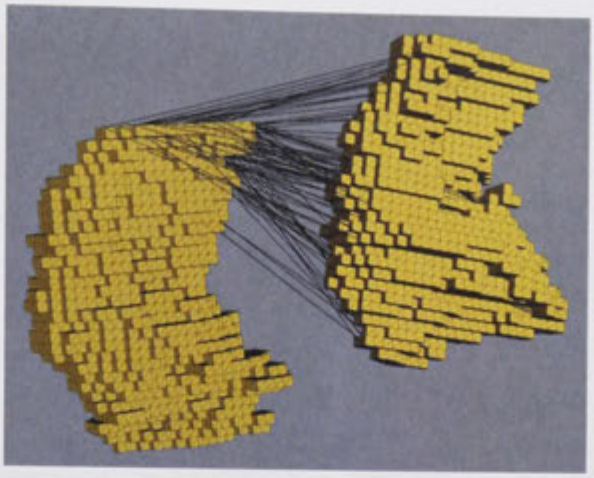


Figure 8.9: Matching of two real hippocampi 118 and 124 from PATH with no clamping.

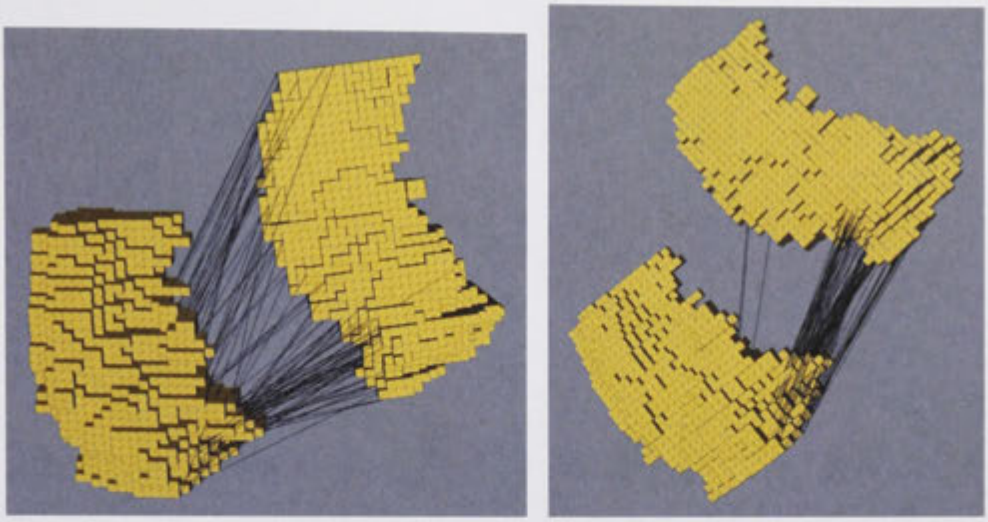
8.3.2 Matching with Clamping

In this section, we perform experiments of matching with clamping. We demonstrate that by using clamping, the performance of 3D shape matching is increased significantly.

We conduct the matching with clamping experiments as follows. We perform the 3D shape matching with and without clamping between a same real hippocampus pair named 120 with 2500 nodes and 219 with 2594 nodes from PATH. Firstly, we perform the 3D shape matching without clamping. Secondly, we perform the 3D shape matching with iterating once by using clamping. Finally, we perform the 3D shape matching with two to eight iterations by using clamping.

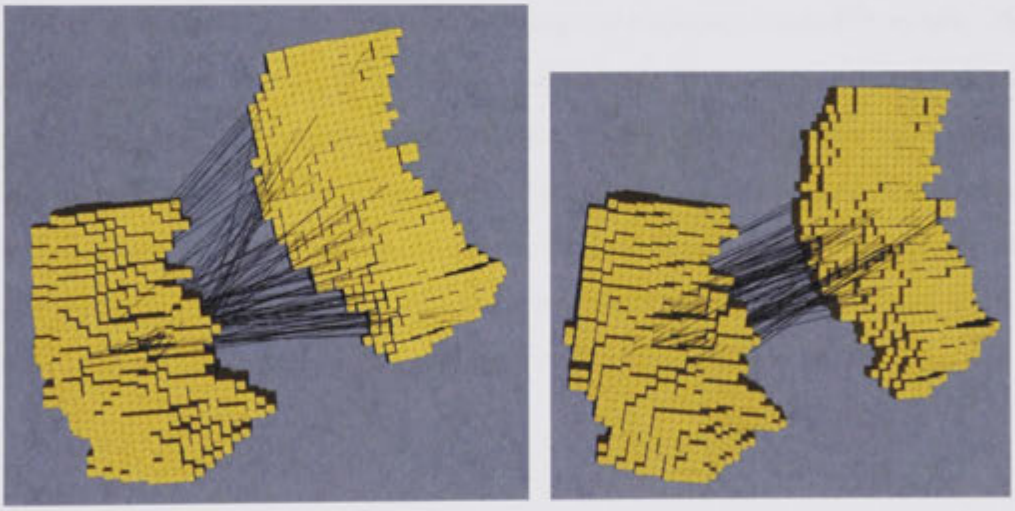
The matching results for the hippocampus head, body, and tail without using clamping are shown in Figures 8.10 (a), 8.11 (a), and 8.12 (a). The correct matching rate is 15%. From the figures we can see that there are many mismatches in all three portions of head, body, and tail.

The matching results after iterating once using clamping are shown in Figures 8.10 (b), 8.11 (b), and 8.12 (b). The correct correspondence matching rate is increased from 15% to 85%. The figures show that there are significantly more consistent matches in all three portions of hippocampus head, body, and tail than without clamping.



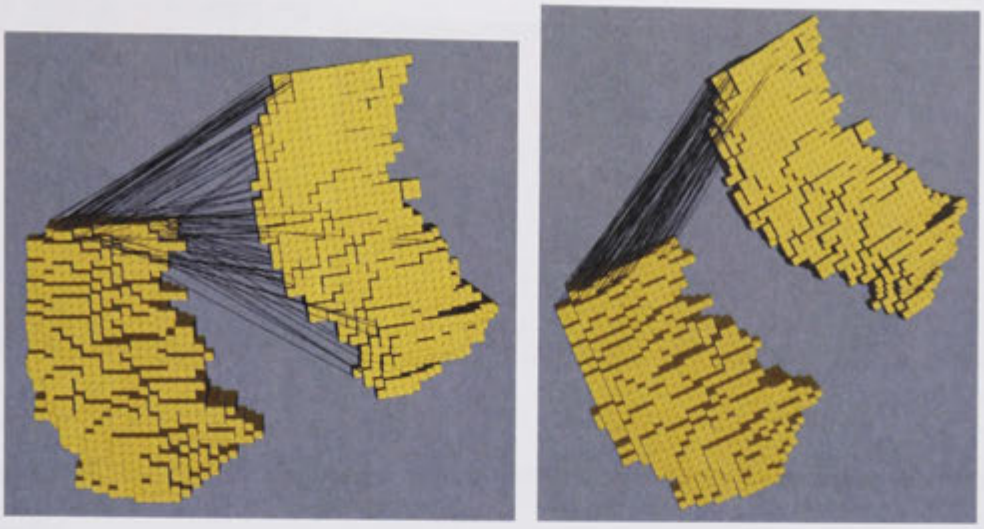
(a) Without clamping. (b) After one iteration of clamping.

Figure 8.10: Comparison of without clamping and after one iteration of clamping for matching of hippocampus head between 120 and 219 from PATH.



(a) Without clamping. (b) After one iteration of clamping.

Figure 8.11: Comparison of without clamping and after one iteration of clamping for matching of hippocampus body between 120 and 219 from PATH.

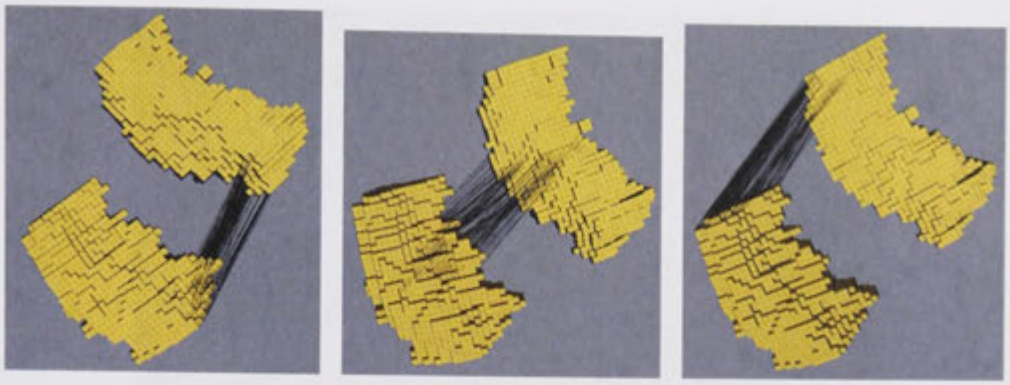


(a) Without clamping.

(b) After one iteration of clamping.

Figure 8.12: Comparison of without clamping and after one iteration of clamping for matching of hippocampus tail between 120 and 219 from PATH.

After we run iteration two, the correct matching rate is increased from 85% to 95%. The subsequent iterations only slightly increase the matching rate. The maximum correct matching rate we obtain is 97% after running eight iterations. The final matching results for hippocampus head, body, and tail are shown in Figures 8.13 (a), (b), and (c). The figures demonstrate that more consistent matches are obtained by running more iterations by using clamping. For majority of point correspondences, head matches to head, body matches to body, and tail matches to tail. Lines are also largely parallel, showing a topologically consistent match.



(a) Head. (b) Body. (c) Tail.

Figure 8.13: Matching of hippocampus between 120 and 219 from PATH after eight iterations of clamping.

We have analysed the outlier distribution of correspondences after using clamping. As shown in Figures 8.14 and 8.15, the outliers are concentrated in the corners of the hippocampus. The hippocampus in the figures is 120 from PATH. The correspondences are established by running clamping eight times between 120 and 219 from PATH.

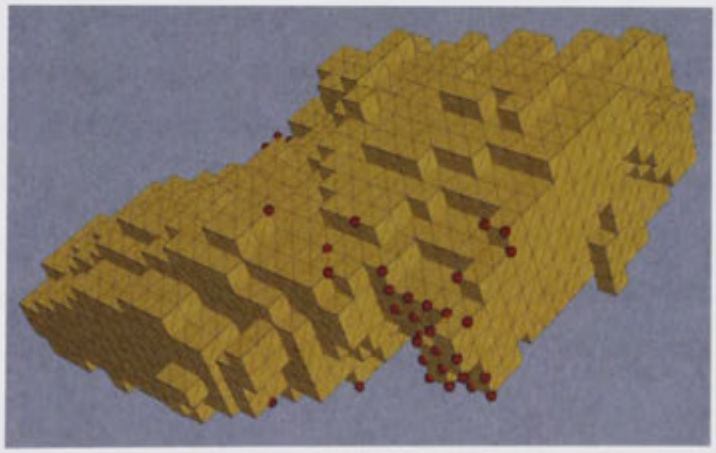


Figure 8.14: Outlier distribution of correspondences after using clamping. The hippocampus is 120 from PATH. The correspondences are established by running clamping eight times between 120 and 219 from PATH.

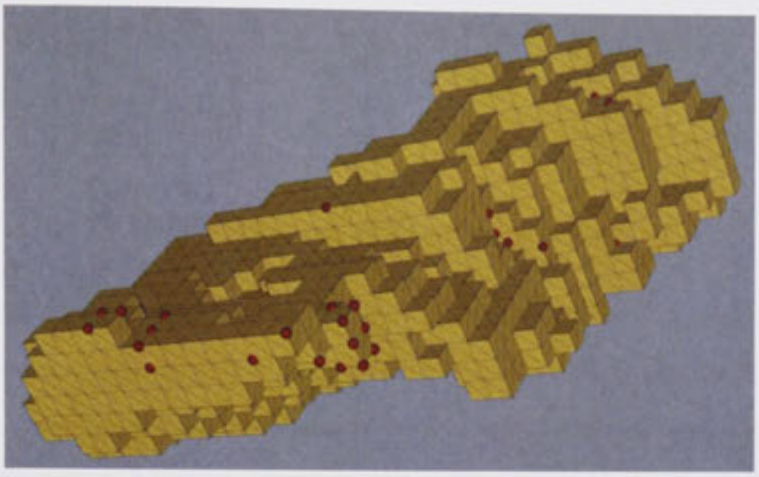


Figure 8.15: Another view of outlier distribution of correspondences after using clamping. The hippocampus is 120 from PATH. The correspondences are established by running clamping eight times between 120 and 219 from PATH.

8.3.2.1 Clamping versus Neighbourhood

We investigate the impact of the immediate neighbours of correct matches on clamping results for 3D shape matching. In order to do this, we performed experiments of 3D shape matching by clamping a correct matching node to its correspondence and all of its immediate neighbours. By clamping to the node's correspondence's immediate neighbours, the correct matching rate can be increased by 2%. Since clamping a correct matching node to its correspondence's immediate neighbours only slightly increases the correct matching rate, in the subsequent experiments, we will not take account of clamping to the node's correspondence's immediate neighbours.

8.4 Rigid Body Registration Experiments

In this section, we perform the experiments of an application of dense correspondence – 3D rigid body registration. The experiments demonstrate the effectiveness of dense correspondence, which forms the basis of non-rigid registration between deformed surfaces. By using

Noise Level	Average Error Rate	Standard Deviation
10%	11.7%	0.0155
50%	39.3%	0.1132
90%	79.3%	0.0554

Table 8.7: Error rate of correspondence matching.

synthetic data, we demonstrate that based on point correspondences obtained by 3D shape matching, we can accurately estimate the translation and rotation between two 3D shapes. For real data, we show 3D rigid body registration between two hippocampi by showing the overlapping results of the two hippocampi after applying the estimated translation and rotation.

8.4.1 Synthetic Data

We use the synthetic data as described in Section 8.1.1 to perform the registration experiments. The experiments are to register the surface mesh with its curvature disturbed counterparts. The surface mesh is shown in Figure 8.1. We run the max product belief propagation as described in Section 5.3 to perform the probabilistic inference.

8.4.1.1 Matching with Curvature Noise

We disturbed the curvature by adding random and uniformly distributed noise with distribution intervals of [-10%,10%], [-50%,50%], and [-90%,90%] of Gaussian curvature at each node respectively. Gaussian curvature is obtained from the estimated principal curvatures. We run the matching 100 times at each noise level. The average error rate and standard deviation at each noise level are summarised in Table 8.7. On average, 88.3% of the correspondences are correct for the noise distribution interval [-10%,10%]; for the noise distribution interval [-50%,50%], the correctness rate is 60.7%; and for the noise distribution interval [-90%,90%], only 20.7% of the correspondences are correct.

8.4.1.2 Verification of 3D Transformation Estimation

In the following, we demonstrate the experiments to estimate the rotation and translation of 3D rigid body registration. In order to demonstrate the transformation results, we have rotated the curvature disturbed surfaces to 90 and 30 degrees respectively along an axis and shifted certain positions for translation. For a global optimal registration, all the shifted translations are estimated accurately. The corresponding estimated rotation matrix for the 90 degree rotated 3D surface from eigenvalue decomposition based approach is,

$$\begin{bmatrix} 0.00360201 & 0.999652 & -0.0261496 \\ -0.99936 & 0.00266769 & -0.0356773 \\ -0.0355952 & 0.0262614 & 0.999021 \end{bmatrix},$$

which is very close to $\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, 90 degree rotation along Z axis.

For the 30 degree rotated 3D surface, the corresponding estimated rotation matrix from eigenvalue decomposition based approach is,

$$\begin{bmatrix} 0.866019 & 0.500011 & -0.00000045 \\ -0.500011 & 0.866019 & 0.00000004 \\ 0.00000041 & 0.00000018 & 1.000 \end{bmatrix},$$

which is very close to $\begin{bmatrix} 0.866 & 0.500 & 0.000 \\ -0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$, 30 degree rotation along Z axis.

The estimates from the techniques based on singular value decomposition and quaternions are almost identical to the ones obtained by using eigenvalue decomposition based approach, (see Chapter 7), so they are not listed here.

The registration takes less than a second on a Intel Pentium(R) 4 CPU 3.20GHz computer.

8.4.1.3 RANSAC Parameters

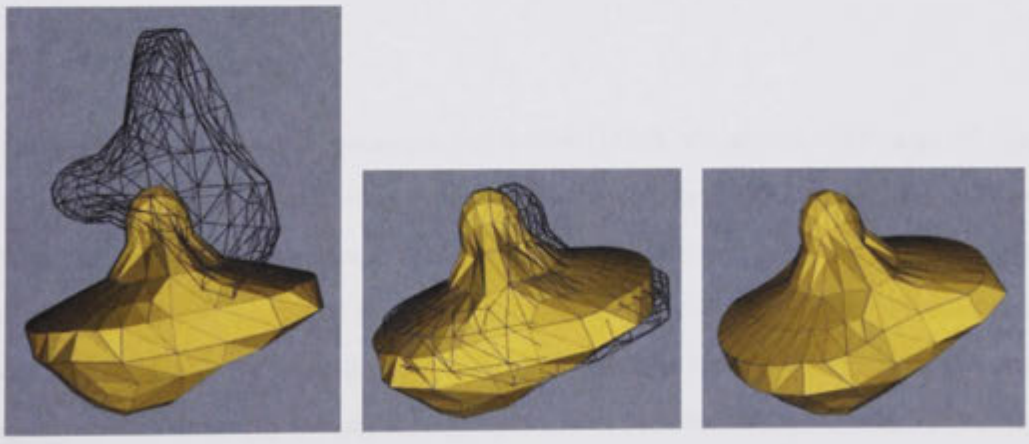
In the simulated data case, according to our experiments, the matching results directly influence the registration accuracy. If more than half of the correspondence matching is correct, then the

Parameters	10% Noise Level		50% Noise Level		90% Noise Level	
τ	0.01	0.10	0.01	0.30	0.01	0.15
ω	190	120	150	100	45	45
Registration	G[1]	G	G	CL[2]	G	CL

Table 8.8: Choice of τ and ω. [1]G: Global optimal [2]CL: Converge to Local sometimes.

rigid body registration always converges to the global optimal solution with some reasonable choice of the Euclidean distance threshold τ and the inlier support number ω. If the correctness rate of the correspondence matching is less than half, then the registration result is somewhat sensitive to the choice of parameter τ and ω. Figure 8.16 (a) shows two meshed surfaces before registration. As shown in Figure 8.16 (b), an inappropriate choice of τ and ω can lead to finding an incorrect solution. A global optimal registration result is shown in Figure 8.16 (c).

From our experiments, a good choice of ω would be close to but less than the number of the correct correspondences. A value which is near but greater than the maximum of the Euclidean distances of the correct matchings would be a good choice for τ. We have performed a series of experiments under different τ and ω. The experiments are summarised in Table 8.8.



(a) Two meshed surfaces before registration.

(b) Registration converged to a local minimum.

(c) Global optimal registration.

Figure 8.16: Registration of synthetic data.

8.4.2 Real Data

In this section, we present results for rigid body registration based on the OASIS and PATH data sets. We perform 3D shape matching based on MRF sparse loopy belief propagation, followed by RANSAC based 3D rigid body registration. After this, we perform matching and registration by iteratively alternating MRF with clamping and RANSAC based 3D rigid body registration.

8.4.2.1 Parameters and Metrics

Let us define the Jaccard index first. Suppose A and B are the volume as expressed as a set of voxels of two hippocampi, the Jaccard index J is calculated as

$$J = \frac{|A \cap B|}{|A \cup B|}.$$

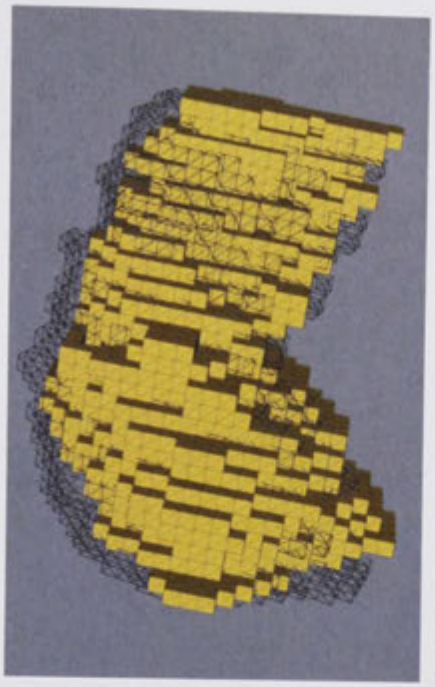
For the experimental conditions, we set the spatial threshold τ as 6 mm for belief propagation without and with clamping, and inlier support set number ω ranging from 300 to 460 points for belief propagation without clamping. The ω will be high for belief propagation with clamping, typically ranging from 1800 to 2300 points.

8.4.2.2 OASIS Data Set

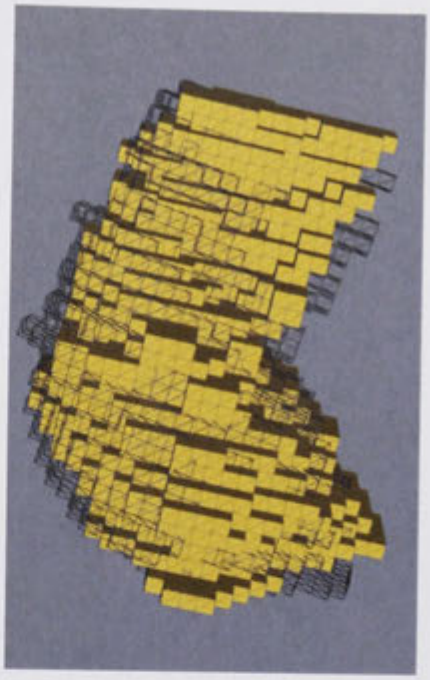
The first part of the human hippocampus data is from OASIS. We perform MRF based 3D rigid body registration experiments using the real human hippocampus pairs. In work that does not form part of this thesis, we have available registrations of the same pairs using the first order ellipsoid (FOE) template based alignment as in [24], and we compare our MRF based method and FOE method. The work is made available by Paulette Lieby. We run 23 hippocampi against another arbitrarily selected hippocampus named 0135 for the experiments. MRF based registration is performed in two stages: firstly without clamping; secondly with two iterations of clamping. We report results after each stage.

After MRF matching without clamping, all of the 23 pairs achieve visually good registration results. In terms of Jaccard index, in 10 of the 23 pairs of registration, MRF based

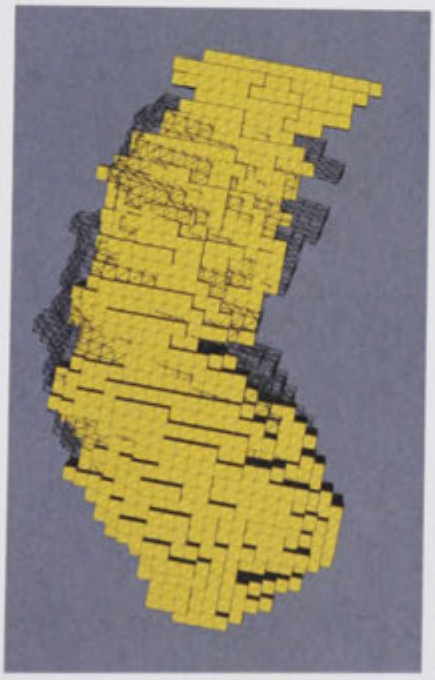
approach performs better; for the rest, FOE template approach performs better. The overall average Jaccard index is very close, MRF 0.6852, and FOE template 0.6927. Two examples of registered pairs where MRF performs better in terms of Jaccard index are shown in Figure 8.17, and two pairs for which FOE template performs better in terms of Jaccard index are demonstrated in Figure 8.18.



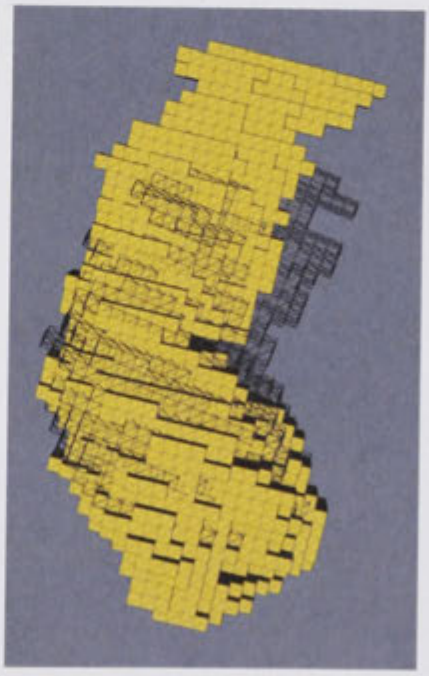
(a) MRF based approach, 0070 vs 0135 (Jaccard index: 0.719849).



(b) FOE based approach, 0070 vs 0135 (Jaccard index: 0.694648).

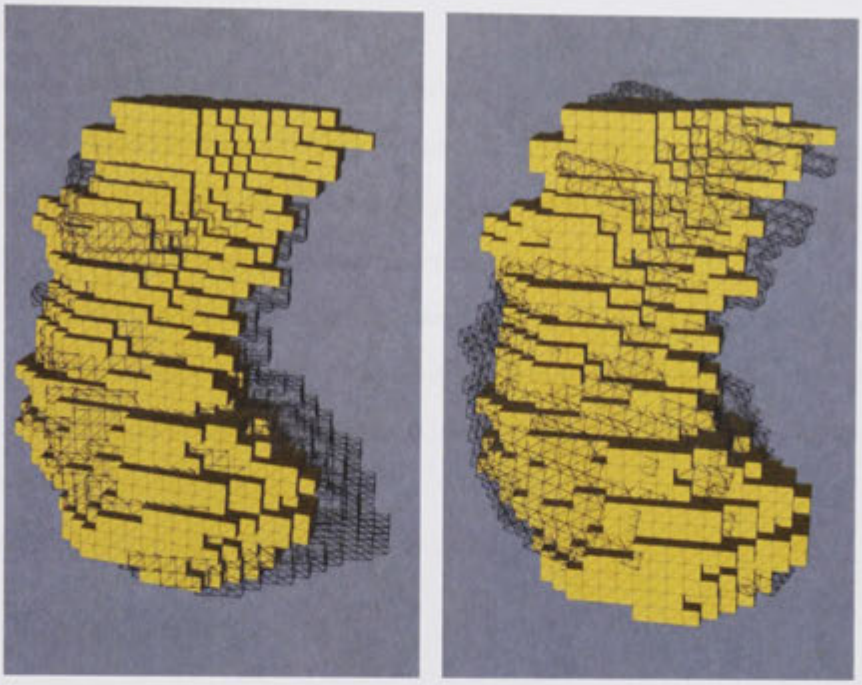


(c) MRF based approach, 0114 vs 0135 (Jaccard index: 0.608843).



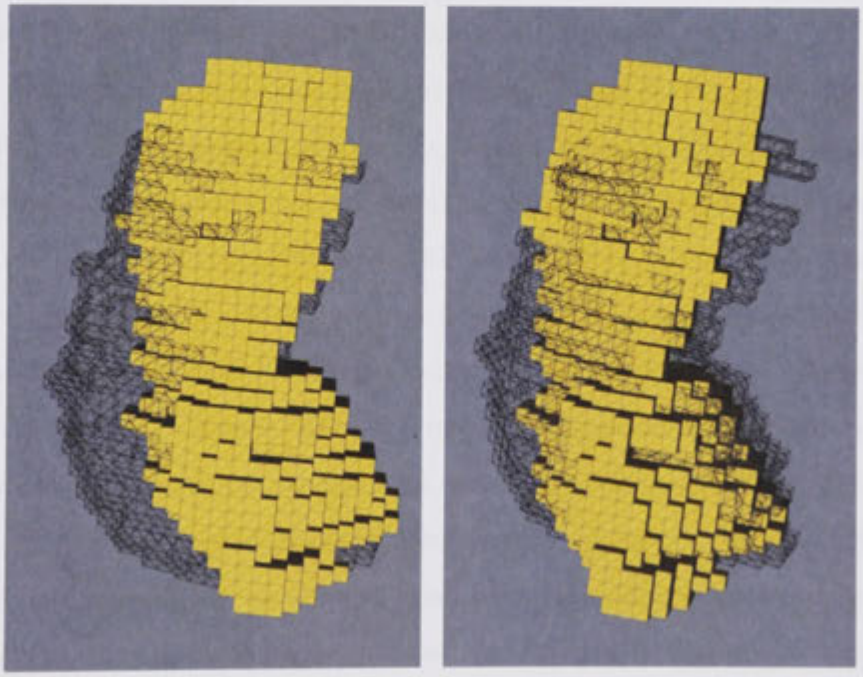
(d) FOE based approach, 0114 vs 0135 (Jaccard index: 0.606888).

Figure 8.17: An example where MRF based registration without clamping performs better in terms of Jaccard index for OASIS data. The wireframe hippocampus is 0135. We do not perform the size normalisation. From (c) and (d) we can see that one hippocampus is bigger than the other.



(a) MRF based approach, 0293 vs 0135 (Jaccard index: 0.629212).

(b) FOE based approach, 0293 vs 0135 (Jaccard index: 0.701916).



(c) MRF based approach, 0457 vs 0135 (Jaccard index: 0.613822).

(d) FOE based approach, 0457 vs 0135 (Jaccard index: 0.630242).

Figure 8.18: An example where FOE template based registration performs better in terms of Jaccard index for OASIS data. The wireframe hippocampus is 0135. The MRF based approach is without clamping.

Clamping

After running clamping twice, the overall average Jaccard index is increased to 0.6966 for MRF, which is slightly better than FOE template 0.6927. In terms of Jaccard index, the number of registered pairs for which the MRF based approach performs better is increased to 12. For the other 11 cases, FOE template approach performs better.

The registration results of the 23 hippocampi against hippocampus named 0135 before clamping are illustrated in Table 8.9. The results after running clamping twice are illustrated in Table 8.10. The comparison of the Jaccard index of MRF before and after clamping twice and FOE is shown in Figure 8.19.

Images showing all registrations appear in appendix.

8.4.2.3 PATH Data Set

Secondly, we use the data from PATH to perform the registration experiments. We randomly select 20 hippocampi from 478 datasets aged 60s against another arbitrarily selected hippocampus named 219 from this set to perform the registration experiments.

The experimental results are summarised as follows. Overall, MRF gives a visually correct registration for 16 of 20 hippocampi. Four hippocampus pairs show some small displacement that looks incorrect as a registration, and we call the cases as "bad". For the four "bad" cases, the average Jaccard index was 0.6458 for MRF and 0.6833 for FOE template. Over the 20 registered pairs, the average Jaccard index was 0.6646 for MRF and 0.6939 for FOE template.

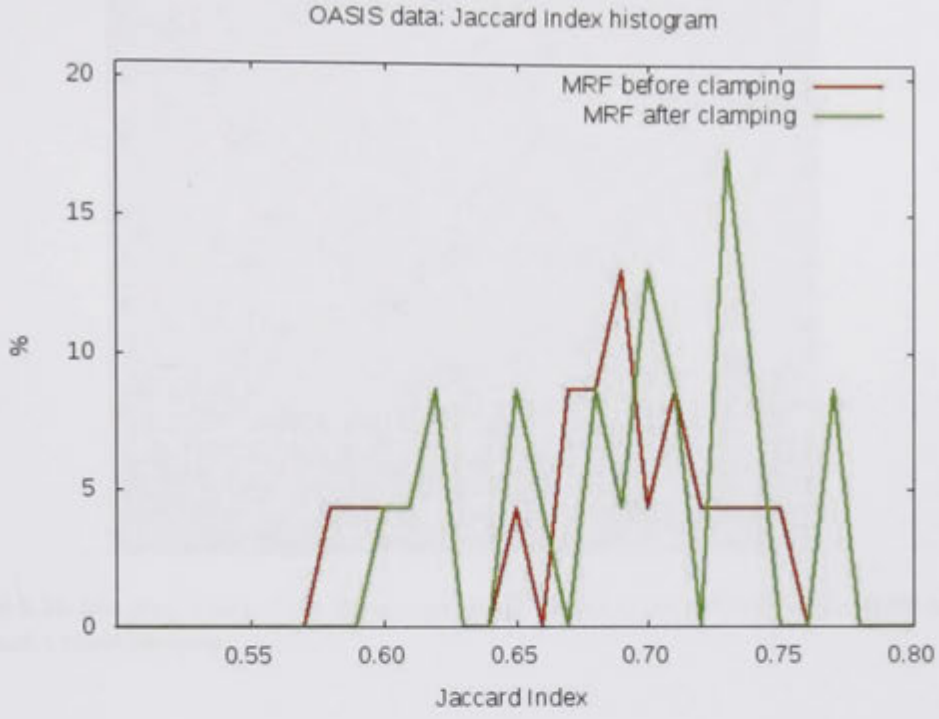
A good registration with MRF between two hippocampi named 118 with 2700 nodes and 124 with 2698 nodes is shown in Figure 8.21, which is based on the matching shown in Figure 8.20. During the experiments, we found one case that is flipped with MRF. After we add sequential information to guide the three point selection sequentially, then the flipped case can be avoided. Specifically, we constrain point choice so that each of three points is chosen from the head, body and tail of a hippocampus respectively. The flipped case between two hippocampi named 496 and 219 is shown in Figure 8.22, and the corresponding corrected case is shown in Figure 8.23. The flipped case occurs mainly because the inlier support set is small.

Hippocampus	Node Number	Inlier Set %	Jaccard Index (MRF)	Jaccard Index (FOE)	Visual Verification	Comparison to FOE
0030	2416	19	0.58027	0.603833	good	better
0068	2486	15	0.654325	0.647277	good	similar
0070	2164	20	0.719849	0.694648	good	similar
0072	2308	19	0.69393	0.695111	good	similar
0078	2404	20	0.770615	0.776554	good	similar
0109	2002	19	0.683851	0.704741	good	similar
0112	2464	21	0.746379	0.730797	good	similar
0114	2778	17	0.608843	0.606888	good	better
0130	2540	17	0.677875	0.711068	good	similar
0133	1826	20	0.709635	0.726046	good	similar
0199	2160	19	0.67658	0.684201	good	similar
0200	2432	19	0.68243	0.680014	good	similar
0292	1982	22	0.694979	0.740016	good	similar
0293	2248	19	0.629212	0.701916	good	worse
0303	1898	21	0.753849	0.744746	good	similar
0322	2266	20	0.730281	0.726225	good	similar
0343	2294	18	0.722252	0.720935	good	similar
0358	2552	15	0.624621	0.629567	good	similar
0422	2158	16	0.599629	0.613673	good	similar
0438	2196	18	0.696911	0.715626	good	similar
0455	2308	20	0.773884	0.747865	good	similar
0456	2520	20	0.715639	0.701015	good	similar
0457	2366	13	0.613822	0.630242	good	worse
Average	2294	19	0.685203	0.692739	good	similar

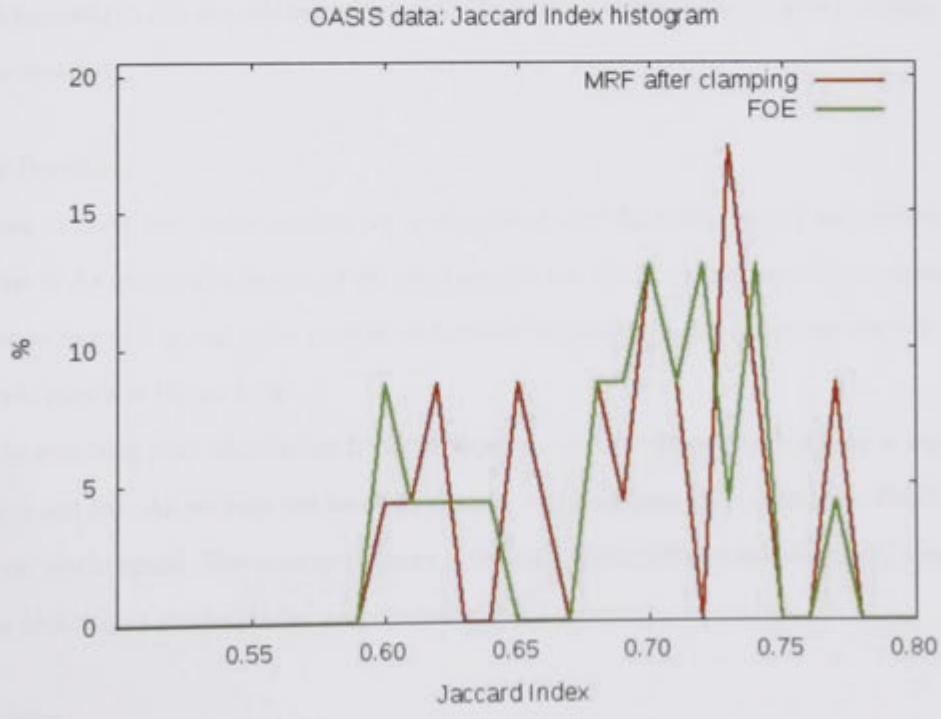
Table 8.9: Registration of 23 hippocampi against another arbitrarily selected hippocampus named 0135 from OASIS before clamping.

Hippocampus	Node Number	Inlier Set %	Jaccard Index (MRF)	Jaccard Index (FOE)	Visual Verification	Comparison to FOE
0030	2416	85	0.620608	0.603833	good	better
0068	2486	91	0.660726	0.647277	good	similar
0070	2164	90	0.746745	0.694648	good	similar
0072	2308	93	0.708202	0.695111	good	similar
0078	2404	94	0.779166	0.776554	good	similar
0109	2002	90	0.685603	0.704741	good	similar
0112	2464	91	0.731412	0.730797	good	similar
0114	2778	85	0.604053	0.606888	good	better
0130	2540	93	0.698172	0.711068	good	similar
0133	1826	93	0.708663	0.726046	good	similar
0199	2160	90	0.682246	0.684201	good	similar
0200	2432	90	0.713686	0.680014	good	similar
0292	1982	93	0.732912	0.740016	good	similar
0293	2248	91	0.653399	0.701916	good	worse
0303	1898	92	0.742436	0.744746	good	similar
0322	2266	93	0.73724	0.726225	good	similar
0343	2294	94	0.732447	0.720935	good	similar
0358	2552	90	0.651934	0.629567	good	similar
0422	2158	86	0.610259	0.613673	good	similar
0438	2196	91	0.704253	0.715626	good	similar
0455	2308	91	0.775579	0.747865	good	similar
0456	2520	93	0.715522	0.701015	good	similar
0457	2366	87	0.625921	0.630242	good	worse
Average	2294	91	0.696573	0.692739	good	similar

Table 8.10: Registration of 23 hippocampi against another arbitrarily selected hippocampus named 0135 from OASIS after clamping.



(a) Comparison of before and after clamping twice.



(b) Comparison of after clamping twice and FOE.

Figure 8.19: Comparison of the Jaccard index of MRF before and after clamping twice and FOE.

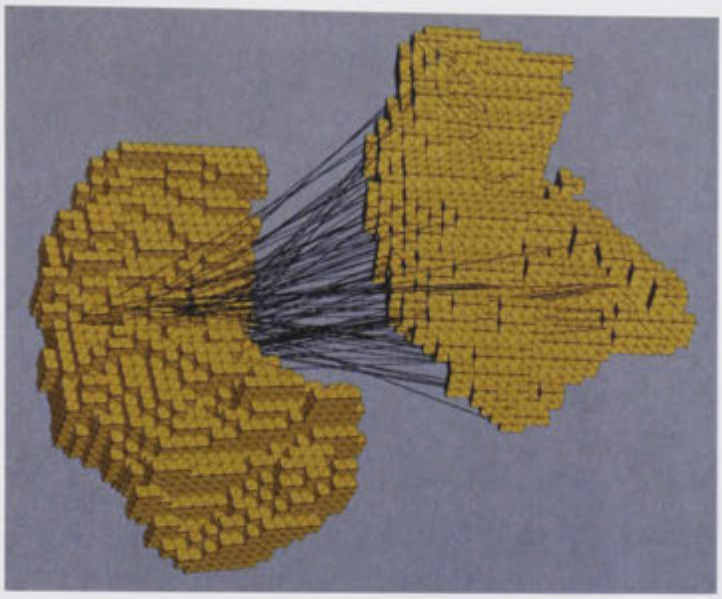


Figure 8.20: Matching results of two hippocampi named 118 and 124 from PATH by using MRF based approach without clamping.

Since sequential information defined by the point selection based on the head, body and tail of a hippocampus can provide the orientation information during registration, the flipped case can be avoided.

Inlier Detection

In order to show how inlier support set is distributed over the whole space, we perform the analysis of the spatial distribution of the inlier support set. The final matches without clamping corresponding to a typical inlier support set between hippocampi named 510 and 219 with 321 inliers is shown in Figure 8.24.

The matching point distribution for inlier support over the whole space is shown in Figures 8.25 (a) and (b). As we may see from the figures, the inlier support is uniformly distributed over the whole space. The correspondences between 510 and 219 are established by running sparse update max product belief propagation without clamping.

Clamping

The clamping technique integrates belief propagation and 3D rigid body registration. It can

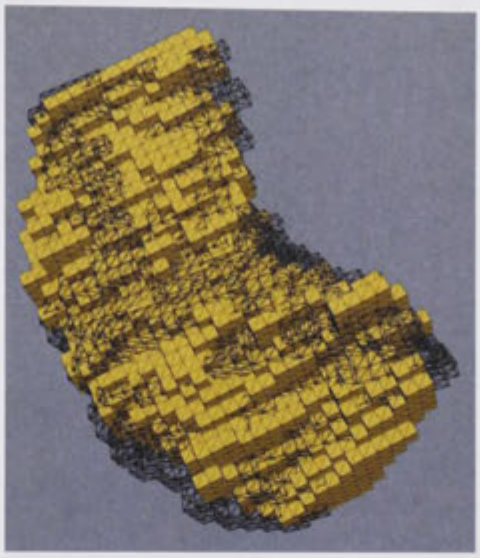


Figure 8.21: A good registration between 118 and 124 from PATH by using MRF based approach without clamping.

Figure 8.22: A flipped case between 496 and 219 from PATH by using MRF based approach without clamping.

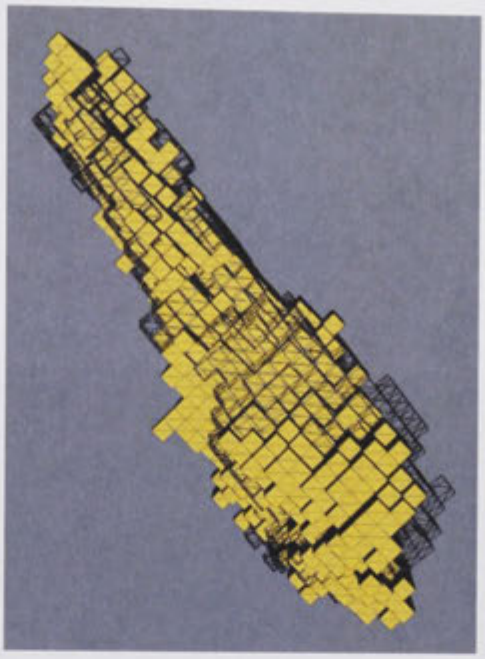


Figure 8.23: The corrected registration of the flipped case between 496 and 219 from PATH by using MRF based approach without clamping.

increase the inlier support set significantly. In a typical case, the proportion of inlier support set can be increased to more than 90% from 13%. A correspondence matching example between hippocampi named 120 and 219 after using clamping twice is shown in Figure 8.26, and the final registration corresponding to the matching is shown in Figure 8.27. From the experiments, the registration based on the matching results using clamping is better than the registration based on matching without clamping. Figure 8.28 shows the comparison of registration with and without clamping.

The average Jaccard index of the 20 registered pairs after clamping twice is increased to 0.6721 for MRF based approach. We show four cases that MRF performs better in terms of Jaccard index after clamping twice in Figures 8.29 and 8.30 compared with FOE template. We have also compared four visually "bad" cases of registration between MRF after clamping twice and FOE template based approaches. The comparison results are shown in Figures 8.31 and 8.32.

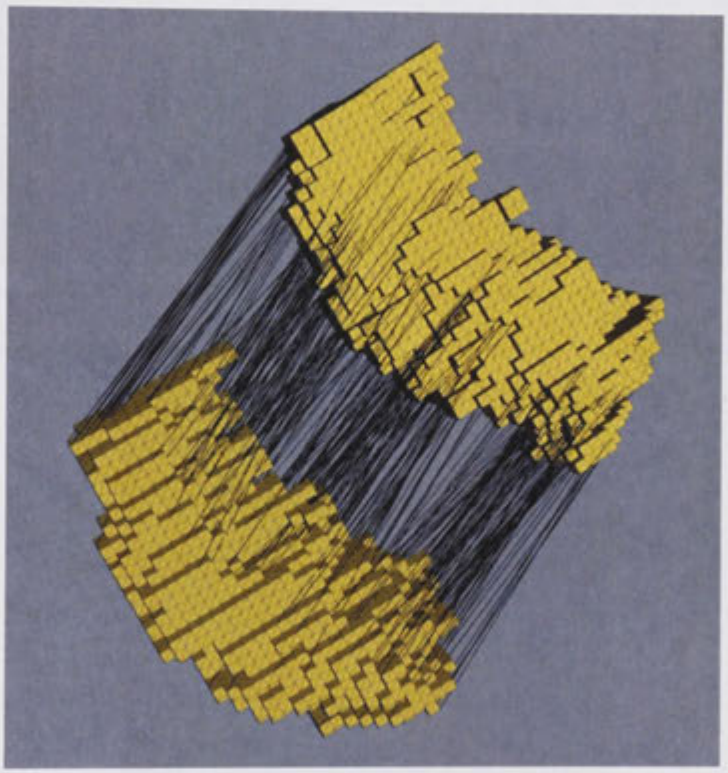
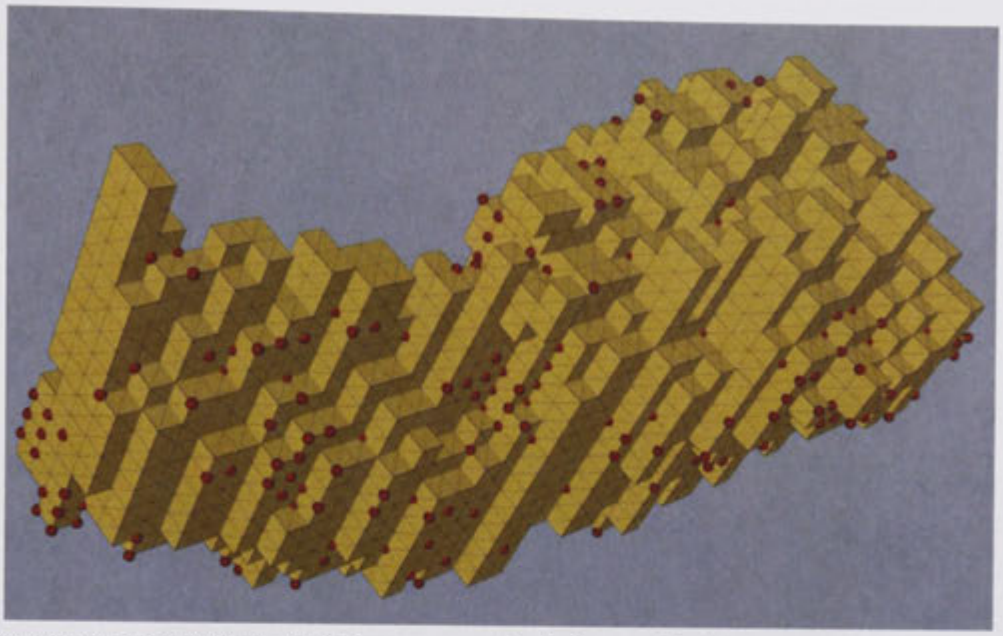


Figure 8.24: The inlier support set matching between 510 and 219 from PATH by using MRF based approach without clamping.



(a) Matching point distribution for inlier support over the whole space. The hippocampus is 510 from PATH. From the figure, the inlier support is uniformly distributed over the whole space. The correspondences between 510 and 219 are established by running sparse update max product belief propagation without clamping.

(b) Another view of matching point distribution for inlier support over the whole space. The hippocampus is 510 from PATH. From the figure, the inlier support is uniformly distributed over the whole space. The correspondences between 510 and 219 are established by running sparse update max product belief propagation without clamping.

Figure 8.25: Matching point distribution.

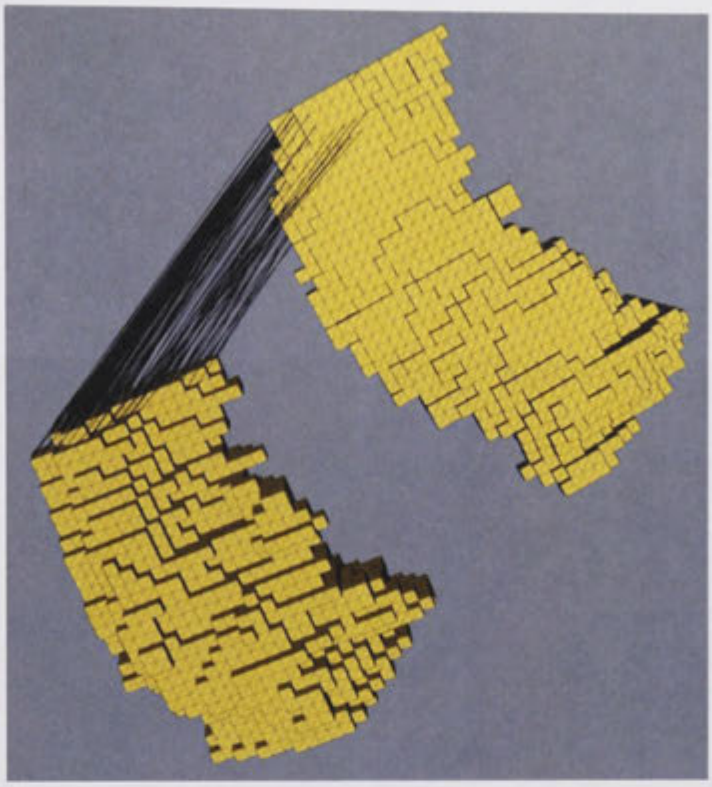


Figure 8.26: Correspondence matching of the hippocampus tail between 120 and 219 from PATH after eight iterations of clamping.

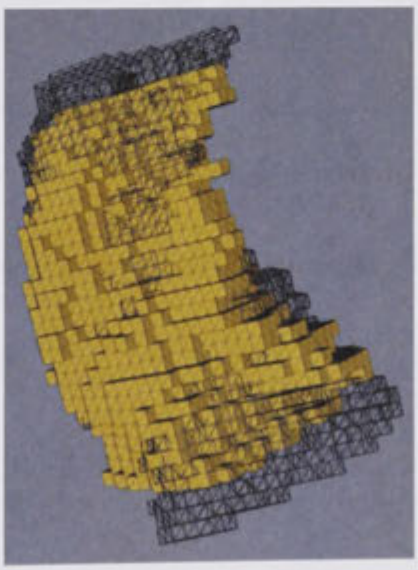
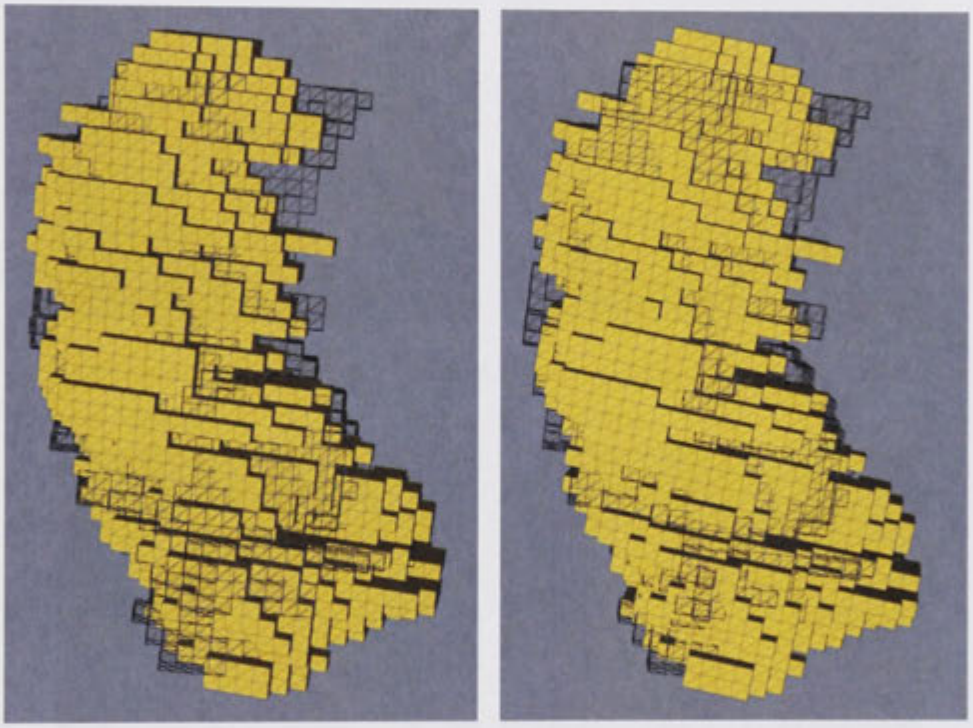


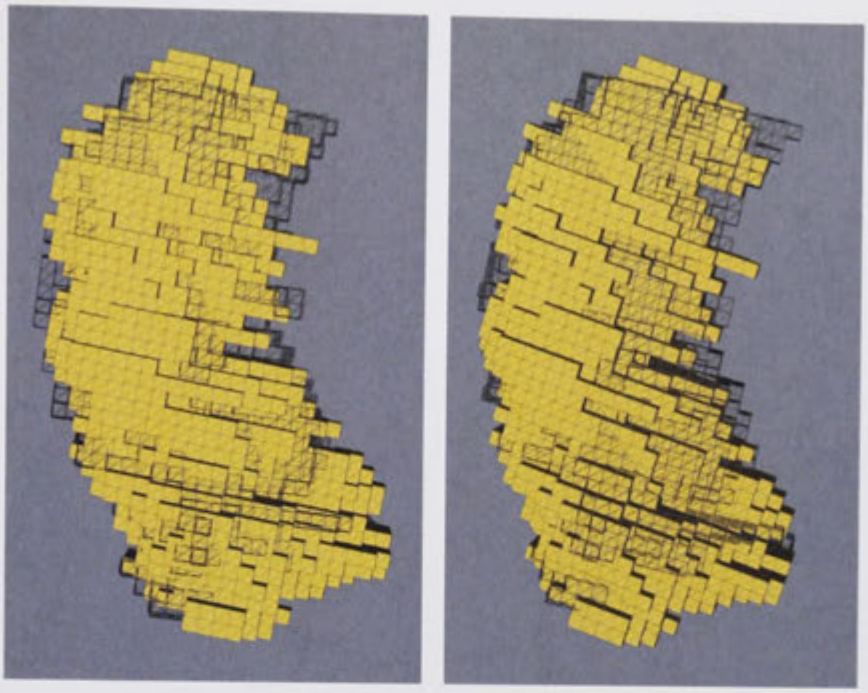
Figure 8.27: Final registration between 120 and 219 from PATH after eight iterations of clamping.



(a) Registration based on the matching without clamping between 107 and 219 from PATH (Jaccard index: 0.687394).

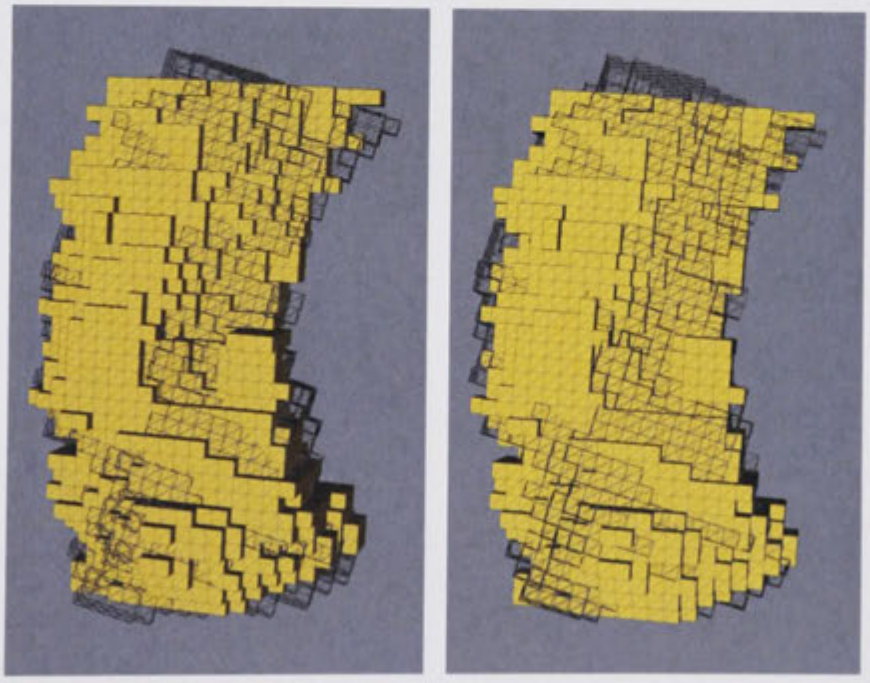
(b) Registration based on the matching using clamping twice between 107 and 219 from PATH (Jaccard index: 0.690612).

Figure 8.28: Comparison of registration with and without clamping between hippocampi 107 and 219 from PATH by using MRF based approach.



(a) MRF based approach, 107 vs 219 (Jaccard index: 0.690612).

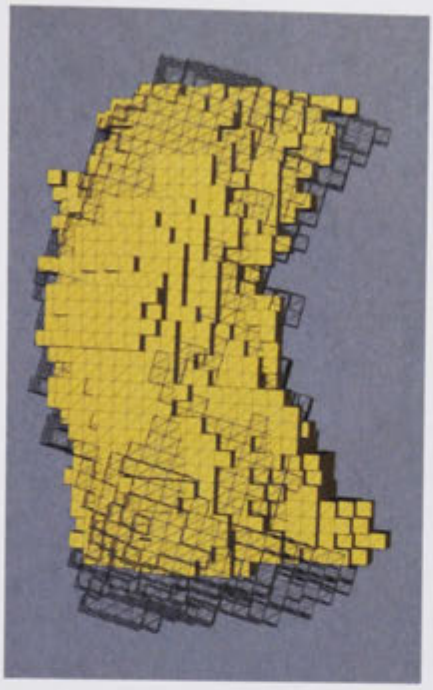
(b) FOE based approach, 107 vs 219 (Jaccard index: 0.685359).



(c) MRF based approach, 335 vs 219 (Jaccard index: 0.73096).

(d) FOE based approach, 335 vs 219 (Jaccard index: 0.726389).

Figure 8.29: Examples where MRF performs better than FOE in terms of Jaccard index after clamping twice cont. The data is from PATH. The wireframe hippocampus is 219.



(a) MRF based approach, 401 vs 219 (Jaccard index: 0.678009).

(b) FOE based approach, 401 vs 219 (Jaccard index: 0.665225).

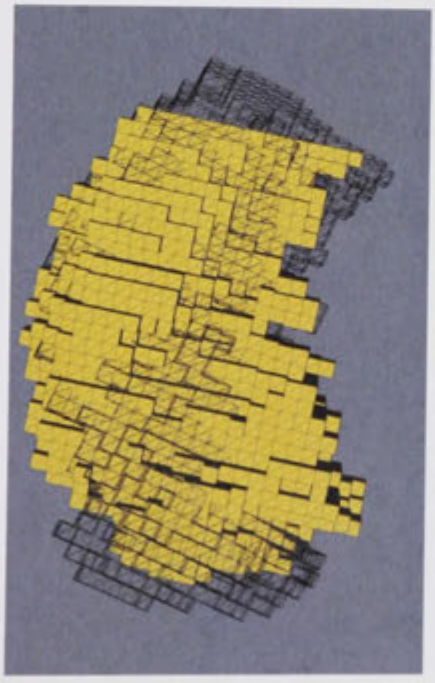
(c) MRF based approach, 427 vs 219 (Jaccard index: 0.716568).

(d) FOE based approach, 427 vs 219 (Jaccard index: 0.714353).

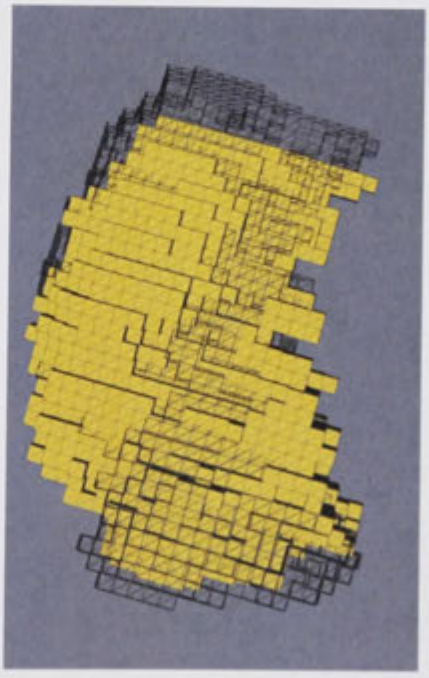
Figure 8.30: Examples where MRF performs better than FOE in terms of Jaccard index after clamping twice. The data is from PATH. The wireframe hippocampus is 219.

(a) MRF based approach, 310 vs 219 (Jaccard index: 0.601434).

(b) FOE based approach, 310 vs 219 (Jaccard index: 0.656712).

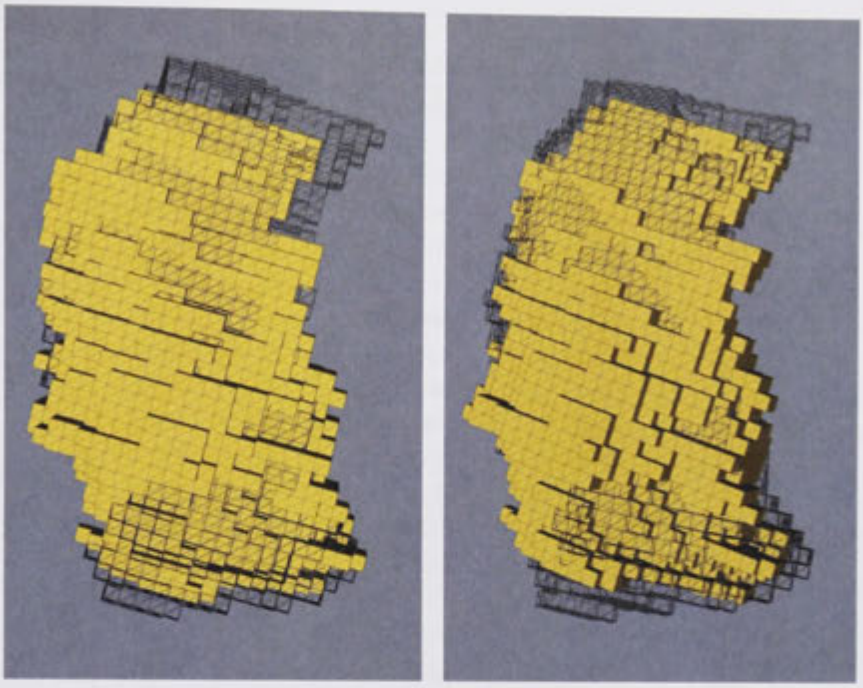


(c) MRF based approach, 548 vs 219 (Jaccard index: 0.615086).



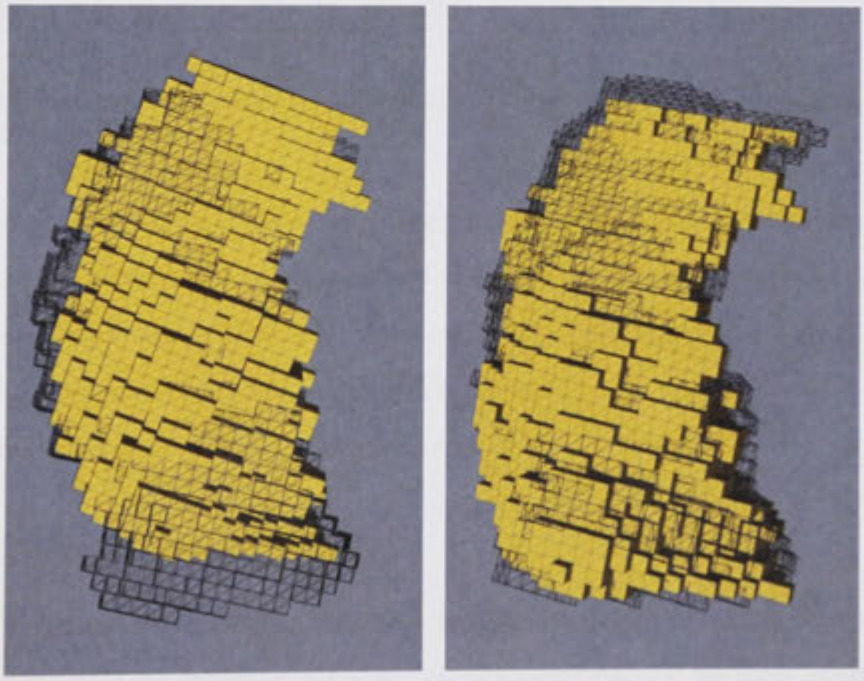
(d) FOE based approach, 548 vs 219 (Jaccard index: 0.654913).

Figure 8.31: Examples of visually "bad" cases of MRF compared to FOE after clamping twice cont. The data is from PATH. The wireframe hippocampus is 219.



(a) MRF based approach, 607 vs 219 (Jaccard index: 0.672824).

(b) FOE based approach, 607 vs 219 (Jaccard index: 0.673002).



(c) MRF based approach, 724 vs 219 (Jaccard index: 0.691951).

(d) FOE based approach, 724 vs 219 (Jaccard index: 0.748613).

Figure 8.32: Examples of visually "bad" cases of MRF compared to FOE after clamping twice. The data is from PATH. The wireframe hippocampus is 219.

Hippocampus	Node Number	Inlier Set %	Jaccard Index (MRF)	Jaccard Index (FOE)	Visual Verification	Comparison to FOE
107	2844	13	0.687394	0.685359	good	worse
120	2500	15	0.654847	0.64692	good	similar
204	2864	12	0.621531	0.628488	good	similar
225	2654	17	0.721569	0.739975	good	similar
310	2618	11	0.622906	0.656712	bad	similar
335	2544	14	0.637512	0.726389	good	similar
351	2488	14	0.605682	0.633006	good	similar
401	2330	13	0.666566	0.665225	good	similar
427	2584	14	0.691988	0.714353	good	similar
488	2520	14	0.669064	0.7206	good	similar
496	2732	11	0.642927	0.67269	good	similar
510	2450	13	0.682353	0.718253	good	similar
548	2540	12	0.629309	0.654913	bad	similar
586	2716	11	0.638847	0.657766	good	similar
607	2864	11	0.63665	0.673002	bad	similar
621	2682	12	0.729982	0.724761	good	similar
640	2806	13	0.674841	0.739008	good	similar
688	2580	15	0.70279	0.777848	good	similar
702	2822	12	0.681291	0.693406	good	similar
724	2446	11	0.694556	0.748613	bad	worse
Average	2629	13	0.664630	0.693864	good	similar

Table 8.11: Registration of 20 hippocampi from PATH against another arbitrarily selected hippocampus named 219 before clamping.

The registration results of the 20 hippocampi against another arbitrarily selected hippocampus named 219 before clamping are illustrated in Table 8.11. The results after running clamping twice are illustrated in Table 8.12. The comparison of the Jaccard index of MRF before and after clamping twice and FOE is shown in Figure 8.33.

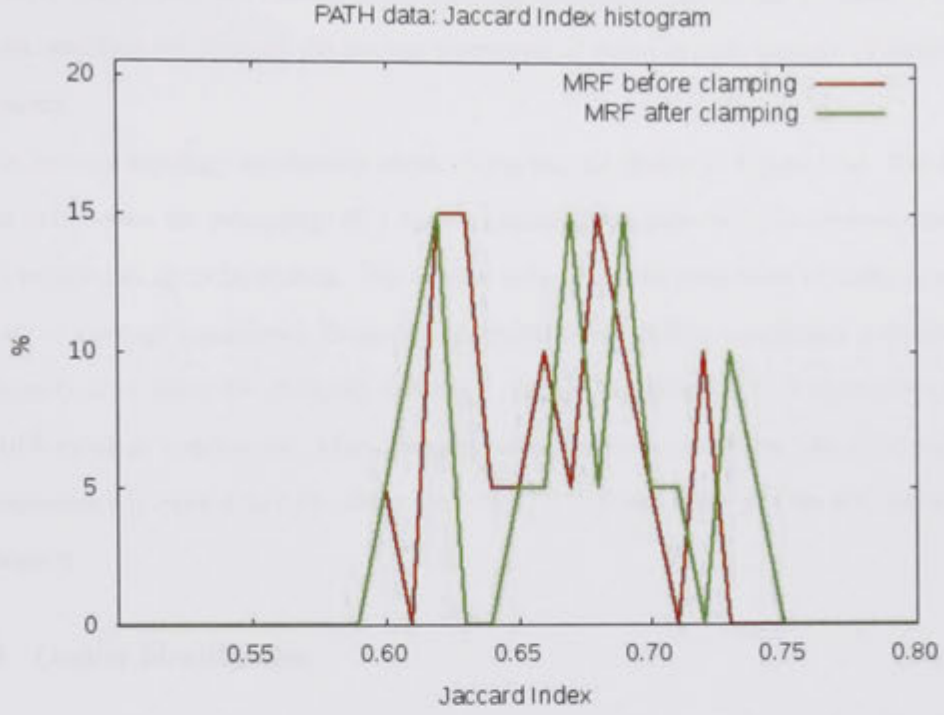
Images showing all registrations appear in appendix.

8.4.3 Topology Consistency Check

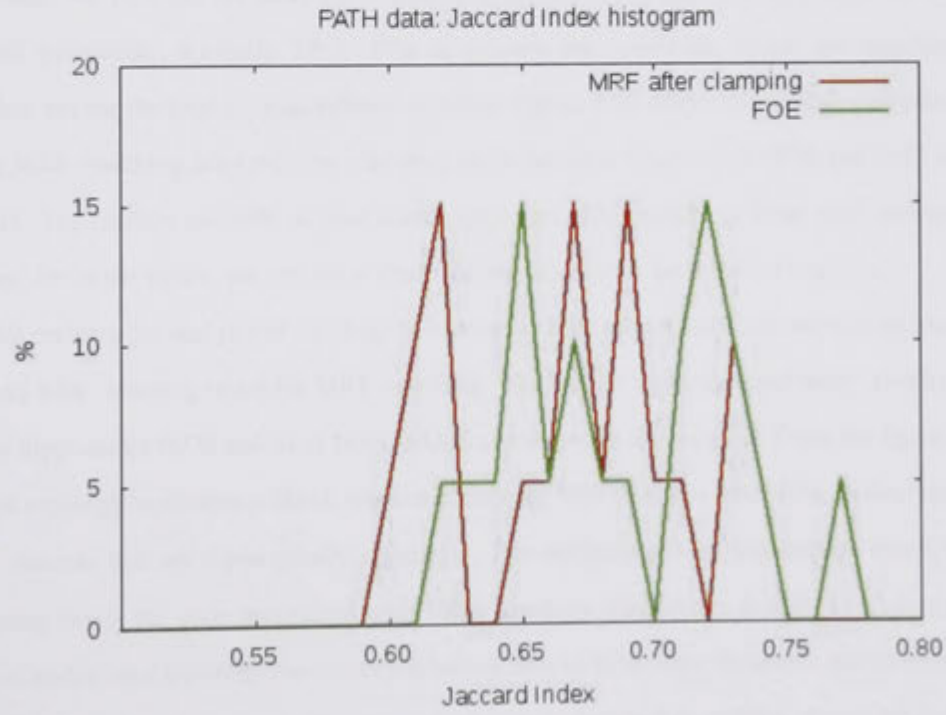
In order to evaluate the registration results, we have run a topology consistency check. Let us define a layer to be the collection of a node's immediate neighbours. We subsequently define layer two as layer's layer, and so on. In the topology consistency check, for a particular node, we check how many of its immediate neighbours are in its correspondence node's neighbours

Hippocampus	Node Number	Inlier Set %	Jaccard Index (MRF)	Jaccard Index (FOE)	Visual Verification	Comparison to FOE
107	2844	90	0.690612	0.685359	good	worse
120	2500	90	0.622274	0.64692	good	similar
204	2864	89	0.62754	0.628488	good	similar
225	2654	92	0.736724	0.739975	good	similar
310	2618	86	0.601434	0.656712	bad	similar
335	2544	92	0.73096	0.726389	good	similar
351	2488	88	0.626832	0.633006	good	similar
401	2330	90	0.678009	0.665225	good	similar
427	2584	89	0.716568	0.714353	good	similar
488	2520	89	0.664045	0.7206	good	similar
496	2732	86	0.611876	0.67269	good	similar
510	2450	90	0.68495	0.718253	good	similar
548	2540	87	0.615086	0.654913	bad	similar
586	2716	90	0.650197	0.657766	good	similar
607	2864	89	0.672824	0.673002	bad	similar
621	2682	89	0.679803	0.724761	good	similar
640	2806	93	0.705988	0.739008	good	similar
688	2580	93	0.742162	0.777848	good	similar
702	2822	90	0.693045	0.693406	good	similar
724	2446	92	0.691951	0.748613	bad	worse
Average	2629	90	0.672144	0.693864	good	similar

Table 8.12: Registration of 20 hippocampi from PATH against another arbitrarily selected hippocampus named 219 after clamping.



(a) Comparison of before and after clamping twice.



(b) Comparison of after clamping twice and FOE.

Figure 8.33: Comparison of the Jaccard index of MRF before and after clamping twice and FOE.

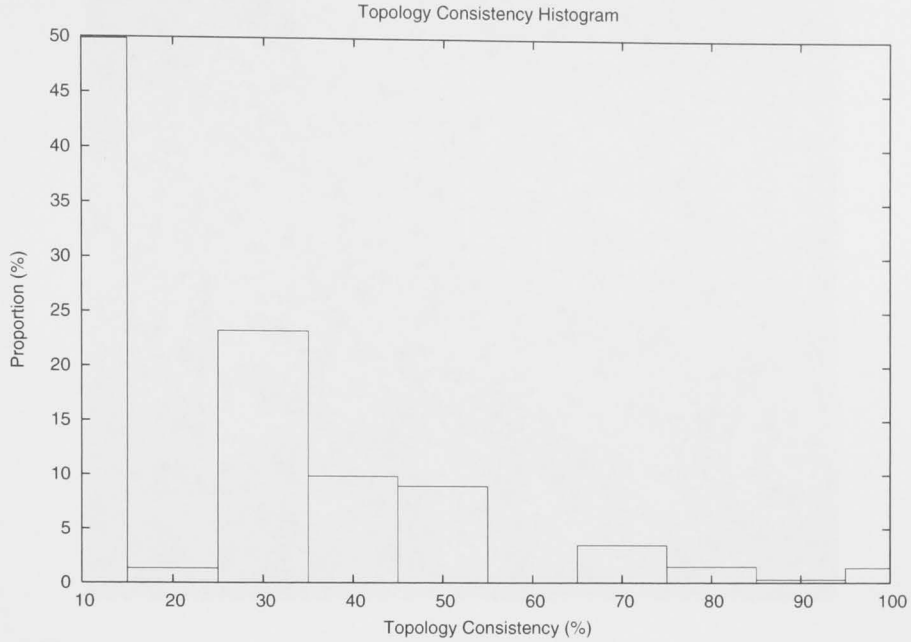
up to layer three. We have calculated the topology consistency for all the 23 cases of OASIS data, and then we obtained the average proportion of nodes in each category of topology consistency.

The average topology consistency check histograms are shown in Figure 8.34. The horizontal axis means the percentage of a node's immediate neighbours in its correspondence node's neighbours up to layer three. The vertical axis means the proportion of nodes in each category of topology consistency. From the experiments, the topology consistency is increased significantly after using the clamping technique. Before clamping, 50% of nodes have less than 10% topology consistency. After clamping twice, the node proportion with 100% topology consistency is more than 40%. There are only 17% of nodes with less than 40% topology consistency.

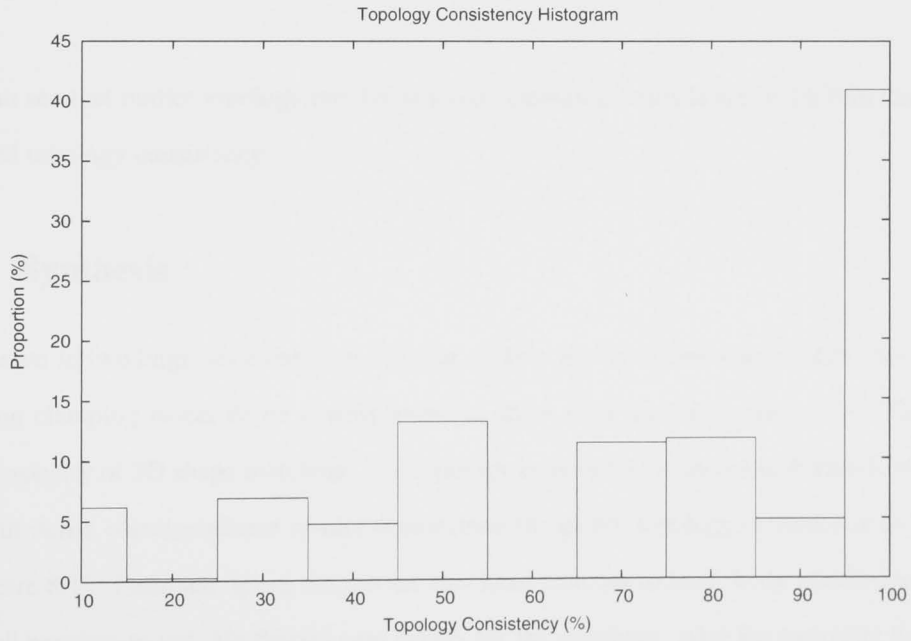
8.4.4 Outlier Identification

In order to check the quality of point-to-point dense correspondences obtained by 3D shape matching, we perform the analysis of outlier identification. After we run clamping twice, a small percentage, normally 5% to 10% of outliers, are identified. These are inconsistent matches among the total correspondence matches. Figure 8.35 shows the outliers identified by using MRF matching after running clamping twice between hippocampi 0070 and 0135 from OASIS. The outliers are 10% of total correspondences, 207 outliers of 2164 total correspondences. From the figure, we can see a small portion of obvious inconsistent matches.

We perform the analysis of topology consistency check of outliers obtained without clamping and with clamping twice for MRF matching. The outlier topology consistency checks between hippocampi 0070 and 0135 from OASIS are shown in Figure 8.36. From the figure, for outlier topology consistency check without clamping, 50% of nodes have 10% of their neighbour matches that are topologically consistent. For outlier topology consistency check with clamping twice, the node proportion with 100% topology consistency is 30%, and more than 60% of nodes have topology consistency between 30% to 80%. This illustrates that in terms of topology the relatively more inconsistent matches are identified as outliers. From the figures



(a) Topology consistency before clamping.



(b) Topology consistency after clamping twice.

Figure 8.34: Comparison of topology consistency before and after clamping. These average topology consistency check histograms are obtained from all the 23 cases of OASIS data. In the topology consistency check, for a particular node, we check how many of its immediate neighbours are in its correspondence node's neighbours up to layer three. The horizontal axis means the percentage of a node's immediate neighbours in its correspondence node's neighbours up to layer three. The vertical axis means the proportion of nodes in each category of topology consistency.

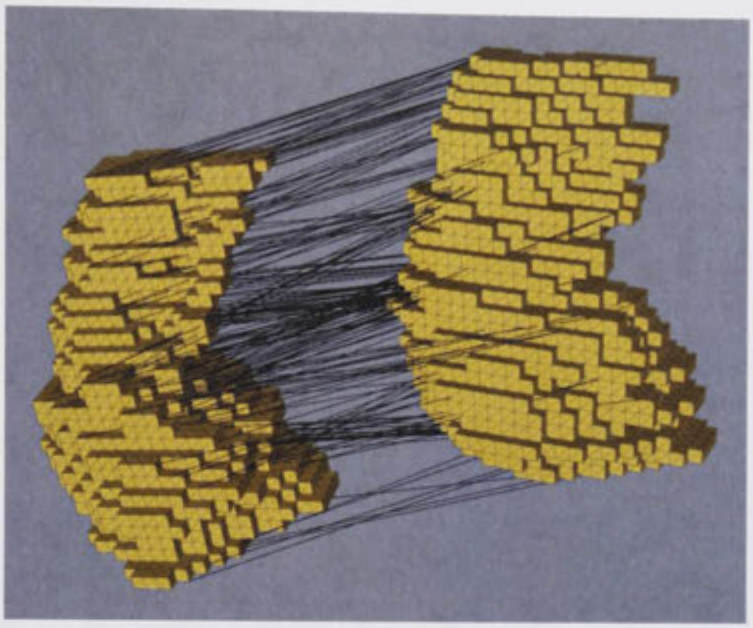


Figure 8.35: Outliers identified by using MRF matching after running clamping twice between hippocampi 0070 and 0135 from OASIS.

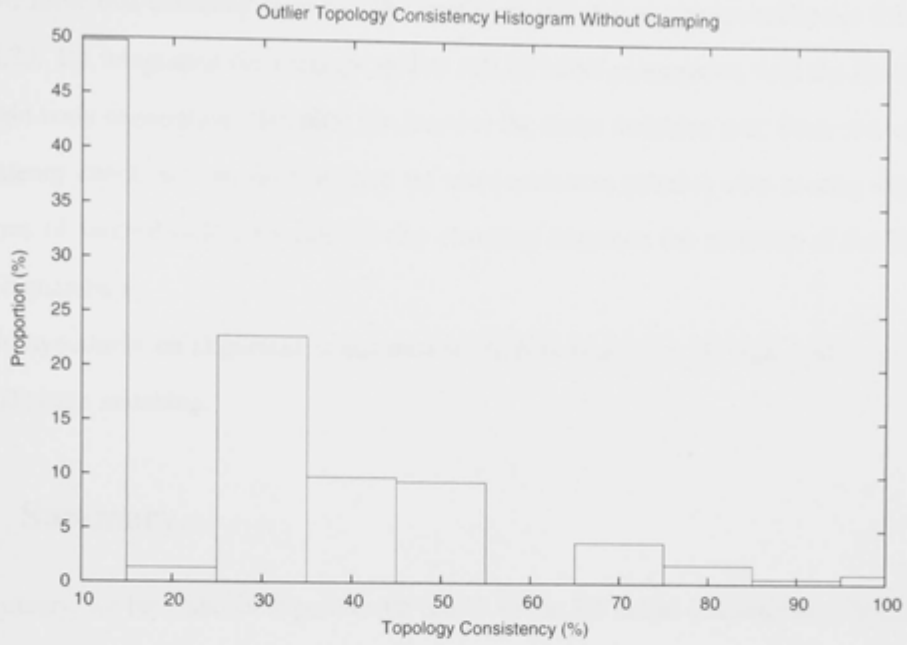
we can see that outlier topology consistency with clamping twice is not much different from general topology consistency.

8.5 Synthesis

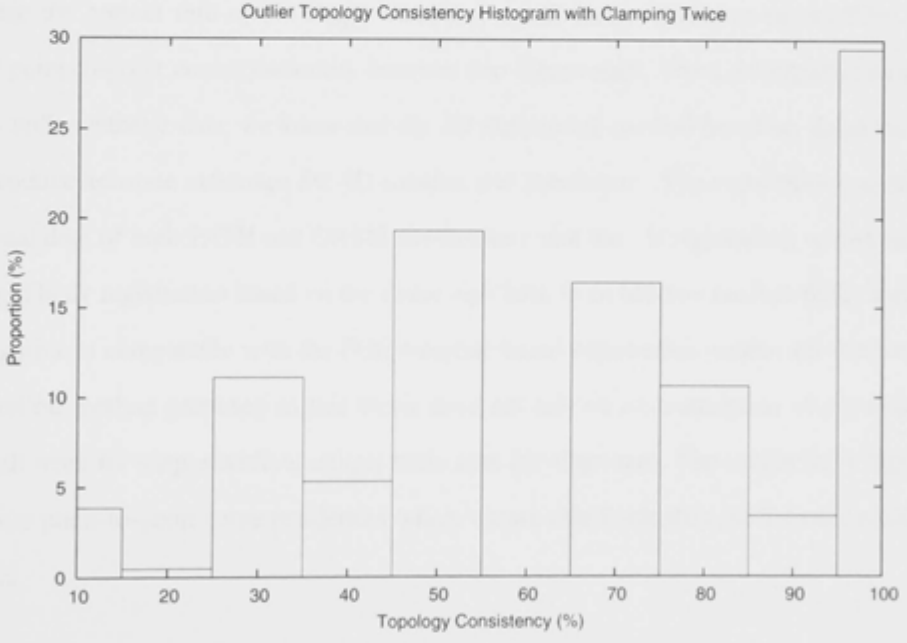
As shown in two large scale data sets, we can reliably recover dense correspondences. After running clamping twice, dense correspondences show high topology consistency. This is a local property of 3D shape matching. The topology consistency is shown in Figure 8.34.

Our dense correspondence results demonstrate the global topology consistency as shown in Figure 8.13. From the figure, we can see that head matches to head, body matches to body, and tail matches to tail. No flipped case occurs for the matching. Also the matching lines are largely parallel supporting the local consistency results.

We have achieved a good quality registration as compared to the first order ellipsoid (FOE) template based alignment. Tables 8.9 and 8.10, and Figures 8.19 (b) and 8.33 (b) show the comparison of registration in terms of Jaccard index.



(a) Outlier topology consistency without clamping.



(b) Outlier topology consistency with clamping twice.

Figure 8.36: Outlier topology consistency checks without clamping and with clamping twice between hippocampi 0070 and 0135 from OASIS. In the topology consistency check, for a particular node, we check how many of its immediate neighbours are in its correspondence node's neighbours up to layer three. The horizontal axis means the percentage of a node's immediate neighbours in its correspondence node's neighbours up to layer three. The vertical axis means the proportion of nodes in each category of topology consistency.

We show that clamping improves the quality of matches as shown in Figures 8.34, 8.19 and 8.33. By integrating the message update rule of belief propagation with the feedback of 3D rigid body registration, clamping can increase the shape matching rate. From the topology consistency check, we can see that there are more consistent matches after running clamping. In terms of Jaccard index, we can see that clamping improves the accuracy of the 3D rigid body registration.

No hypothesis on alignment is assumed while performing the 3D rigid body registration and 3D shape matching.

8.6 Summary

In summary, we have shown experimental results of the 3D shape matching with belief propagation and with clamping. The experimental results show that the clamping technique can increase the correct rate of 3D shape matching significantly. We have successfully shown dense point-to-point correspondences between two hippocampi. From the registration experiments with synthetic data, we know that the 3D registration method based on shape matching can produce accurate estimates for 3D rotation and translation. The registration experiments with real data of both PATH and OASIS demonstrate that the 3D registration results from the 3D rigid body registration based on the shape matching from Markov random fields and belief propagation is comparable with the FOE template based registration results. Furthermore, the registration method proposed in this thesis does not rely on an assumption of pre-alignment and will work for shapes without unique main axes for alignment. The results show high quality dense point-to-point correspondences where we are able to identify inconsistent matches as outliers.

Chapter 9

Conclusion

In this thesis, we have investigated the techniques to recover point based dense correspondence between the surfaces of two neuroanatomical objects. We formulated the 3D shape matching problem with the framework of Markov random fields. Under the framework, the 3D shape matching problem is transformed to an energy function minimisation problem. We have proposed an efficient sparse update loopy belief propagation to minimise the energy function. We have also presented a novel clamping technique to increase the shape matching rate. Based on the outcome of the dense correspondence, we have performed 3D rigid body registration for 3D neuroanatomical objects. For the experiments, we use the hippocampus as a case study for both correspondence matching and registration. The 3D shape matching and registration techniques can be used to facilitate the shape change analysis of general neuroanatomical objects.

9.1 Key Contributions

9.1.1 Recovery of Dense Correspondence

In this thesis, we mainly investigate the recovery of dense correspondence. We formulate the dense correspondence as the outcome of a 3D shape matching problem. We use a global framework by using Markov random fields (MRFs) to formulate the 3D shape matching problem as an energy function minimisation problem. Gaussian curvature is used for similarity measure between nodes of an MRF and its sample instance for the internal energy definition. Pairwise distance defined on 2-element cliques is used to define the external energy. By combining Markov random fields and Gaussian curvature, the dense correspondence can be obtained by

finding the best match between two objects through 3D shape matching.

9.1.2 Need for Loopy Belief Propagation

In order to obtain the solution of the 3D shape matching, we need to solve the energy function minimisation problem. However, solving the minimisation problem is NP hard. We have tried to use randomised algorithms including Gibbs sampler based simulated annealing to minimise the energy function. Experiments show that although the randomised algorithms work well for synthetic data with small state space 200, they do not work for real data with large state space (more than 2000 nodes in this thesis). Therefore, we address this large state space energy function minimisation problem with loopy belief propagation.

9.1.3 Sparse Update Loopy Belief Propagation

When we use loopy belief propagation to minimise the energy function, we face a computational efficiency problem. Each iteration of loopy belief propagation takes too long for real data with large state space (more than 2000 nodes in this thesis). To solve the problem, we propose an efficient belief propagation algorithm – sparse update loopy belief propagation. The sparse update loopy belief propagation is much faster than standard loopy belief propagation by exploiting the topological structure and defining the potential function. Replacing the standard full update with sparse update has little impact on results.

9.1.4 Clamping Technique

In order to enhance the shape matching rate and increase the inlier support set, we propose a novel clamping technique by combining loopy belief propagation message updating rule with the feedback from the 3D rigid body registration. Clamping can significantly increase the correct 3D shape matching rate. Without clamping the correct 3D shape matching rate is around 15%; and after iterating twice by using clamping, the correct 3D shape matching rate is increased to around 90% to 95%. More consistent matches are obtained by running clamping. For majority of point correspondences obtained by performing 3D shape matching using

clamping between two real hippocampi, head matches to head, body matches to body, and tail matches to tail. Lines are also largely parallel, showing topologically consistent matches. Clamping can also enhance the 3D rigid body registration results. The experiments show that by running clamping twice, MRF can achieve better registration results than FOE in terms of the Jaccard index for OASIS data sets.

9.1.5 Dense Correspondence as Non-rigid Registration

By using the clamping technique, we can achieve high rate of correct matchings. The rate of correct matching is more than 90% after clamping. This set of correct matches form a point-to-point dense correspondence between the two matched surfaces, which show topological consistency. Such a dense point-to-point correspondence would form an excellent basis for a non-rigid registration.

9.1.6 An Application of Dense Correspondence

We run an application of dense correspondence – 3D rigid body registration. From the obtained point-to-point dense correspondence, we randomly choose three pairs of correspondence and estimate 3D rigid body transformation from the three pairs of correspondence. By combining with RANSAC technique, we obtain the maximum inlier support set of correspondences. From the inlier support set, we estimate the final 3D rigid body transformation by using least square based optimal transformation estimation techniques based on eigenvalue decomposition, singular value decomposition, and quaternions. Experimental results show that our MRF based registration approach is comparable to a state-of-the-art approach first order ellipsoid (FOE) template based alignment.

9.2 Future Work

From Section 2.3.2.3 we know that point-to-point based dense correspondence can be treated as a basis for non-rigid registration. By combining with non-rigid ICP, the final non-rigid registration can be realized from point-to-point based dense correspondence. Therefore, it

would be good if we could investigate the technique of non-rigid registration by using dense correspondences by our method as a basis for non-rigid ICP.

3D shape matching and registration techniques can be used for shape based classification. By using registration, different shapes can be brought into a common coordinate system to perform comparison. Our technique's reach would be broadened if we could apply our 3D shape matching and registration techniques to shape classification. These techniques could be used as a diagnostic aid, for example when attempting to detect the presence of Alzheimer's disease based on hippocampal shape.

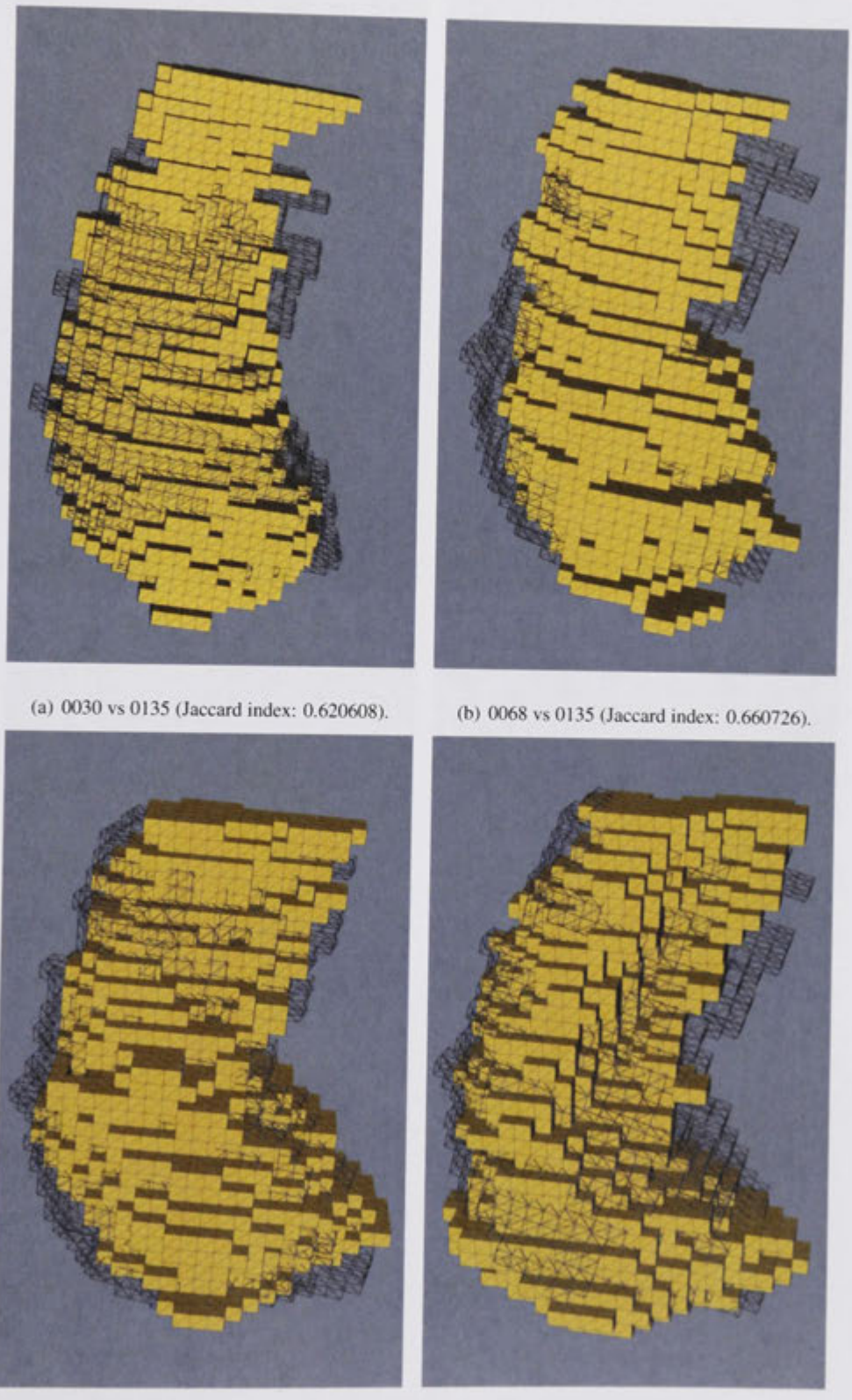
Currently our clamping technique is realized by integrating the message updating rule of belief propagation with the feedback from 3D rigid body registration. We could explore other sources of information as the feedback of the quality of match. For example, we could further investigate the use of the Jaccard coefficient as the quality of match to integrate with message updating rule of belief propagation to realize the clamping.

It would improve our current technique if 3D shape matching were performed using curvature as a vector. The vector contains the multi scale curvature information. In this way, the matching process can incorporate richer geometric information from the surface meshes. By doing this, the shape matching rate is expected to increase.

Appendix

In this appendix, we show the 3D rigid body registration results of OASIS and PATH data sets. The 3D shape matching corresponding to the registration is performed by running clamping twice.

1. OASIS



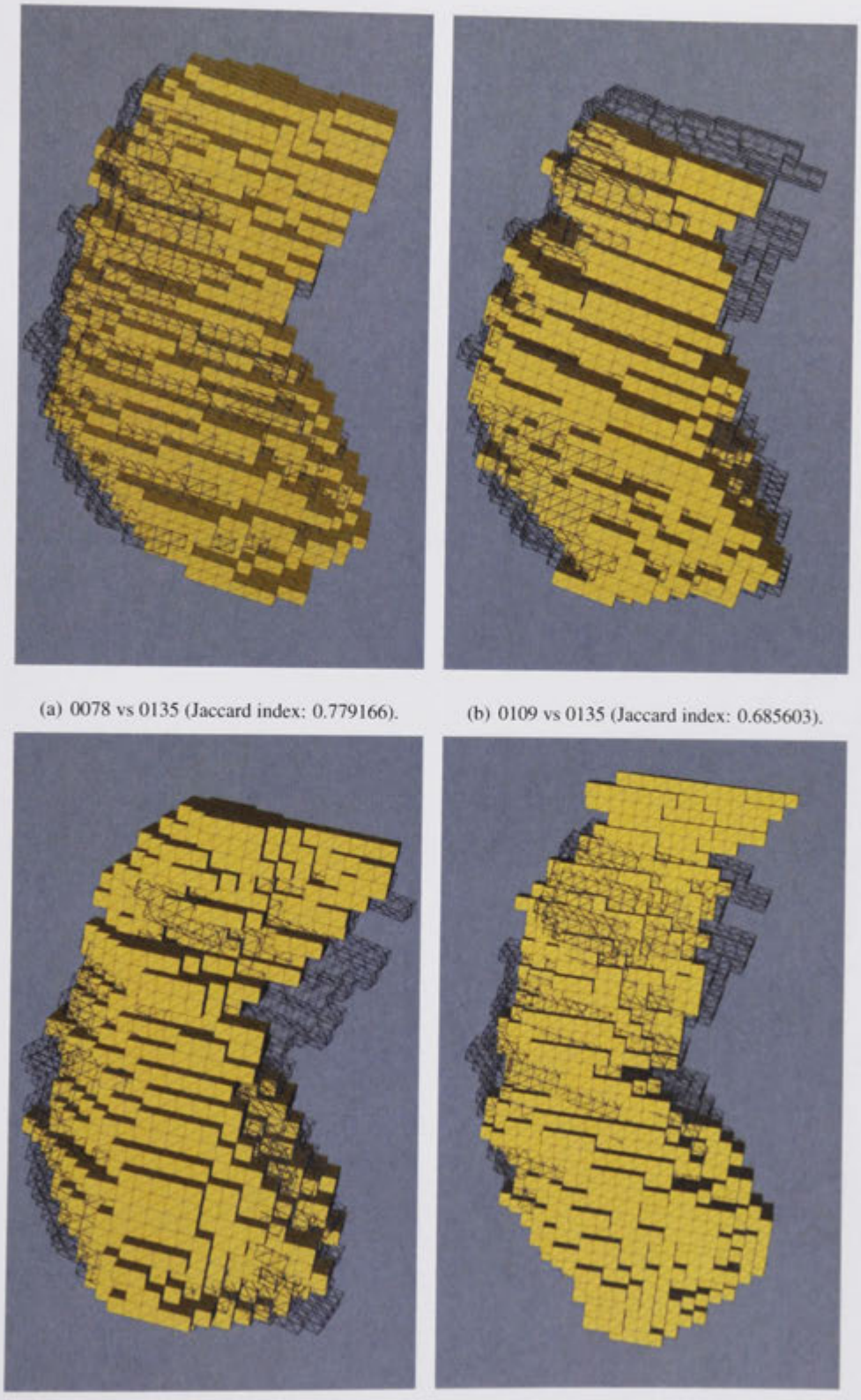
(a) 0030 vs 0135 (Jaccard index: 0.620608).

(b) 0068 vs 0135 (Jaccard index: 0.660726).

(c) 0070 vs 0135 (Jaccard index: 0.746745).

(d) 0072 vs 0135 (Jaccard index: 0.708202).

Figure 1: 3D rigid body registration of OASIS data set. The wireframe hippocampus is 0135.



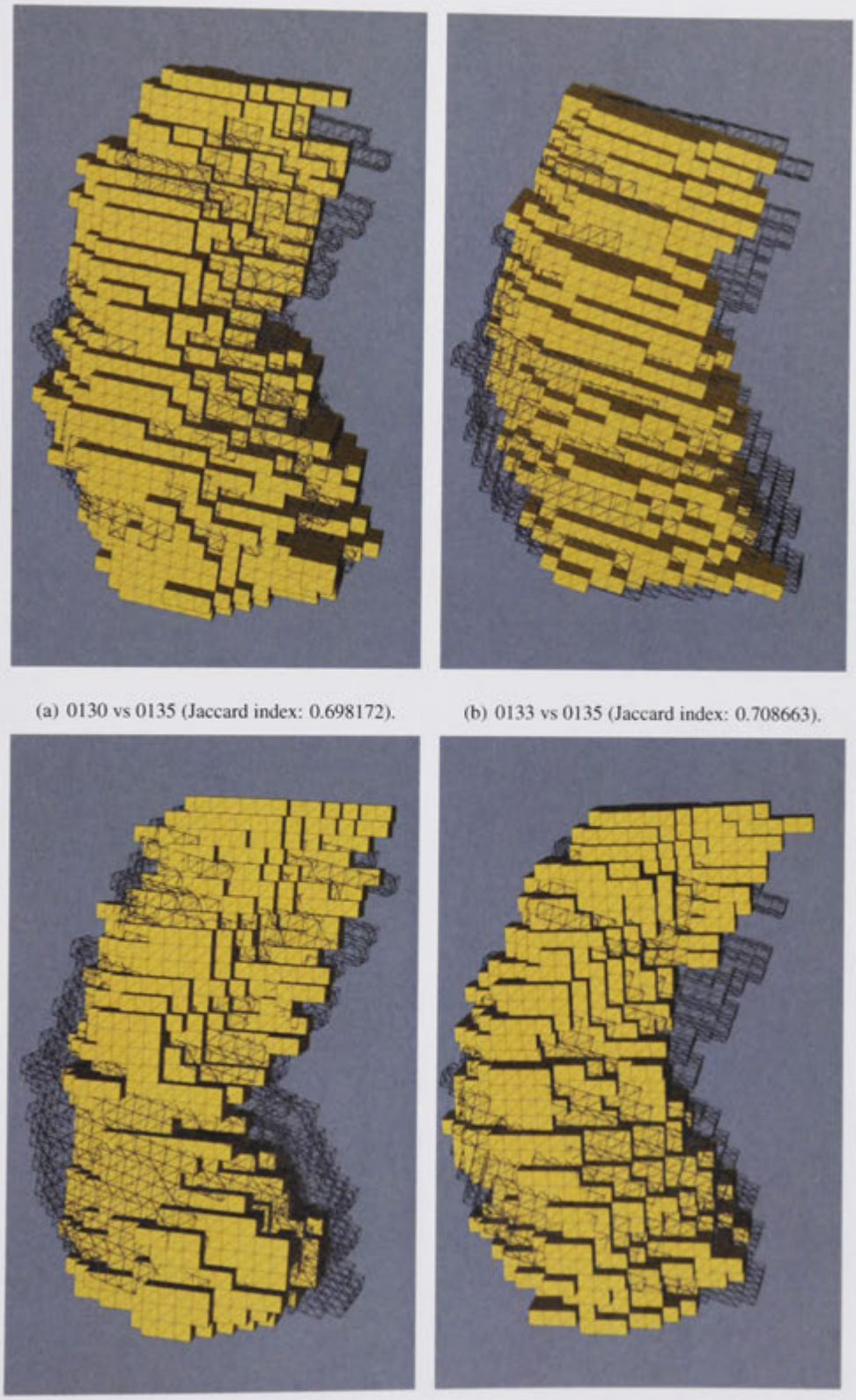
(a) 0078 vs 0135 (Jaccard index: 0.779166).

(b) 0109 vs 0135 (Jaccard index: 0.685603).

(c) 0112 vs 0135 (Jaccard index: 0.731412).

(d) 0114 vs 0135 (Jaccard index: 0.604053).

Figure 2: 3D rigid body registration of OASIS data set. The wireframe hippocampus is 0135.



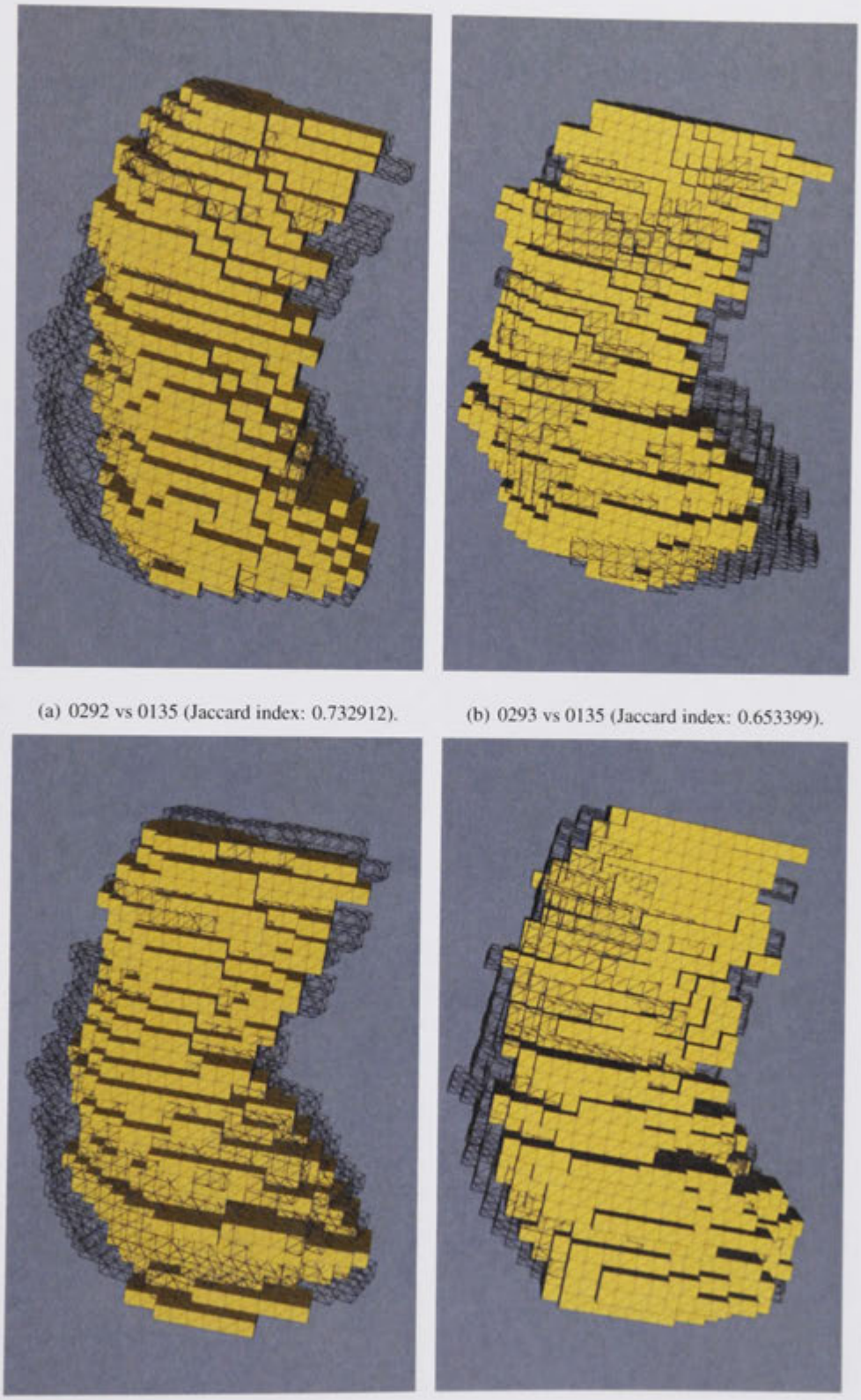
(a) 0130 vs 0135 (Jaccard index: 0.698172).

(b) 0133 vs 0135 (Jaccard index: 0.708663).

(c) 0199 vs 0135 (Jaccard index: 0.682246).

(d) 0200 vs 0135 (Jaccard index: 0.713686).

Figure 3: 3D rigid body registration of OASIS data set. The wireframe hippocampus is 0135.



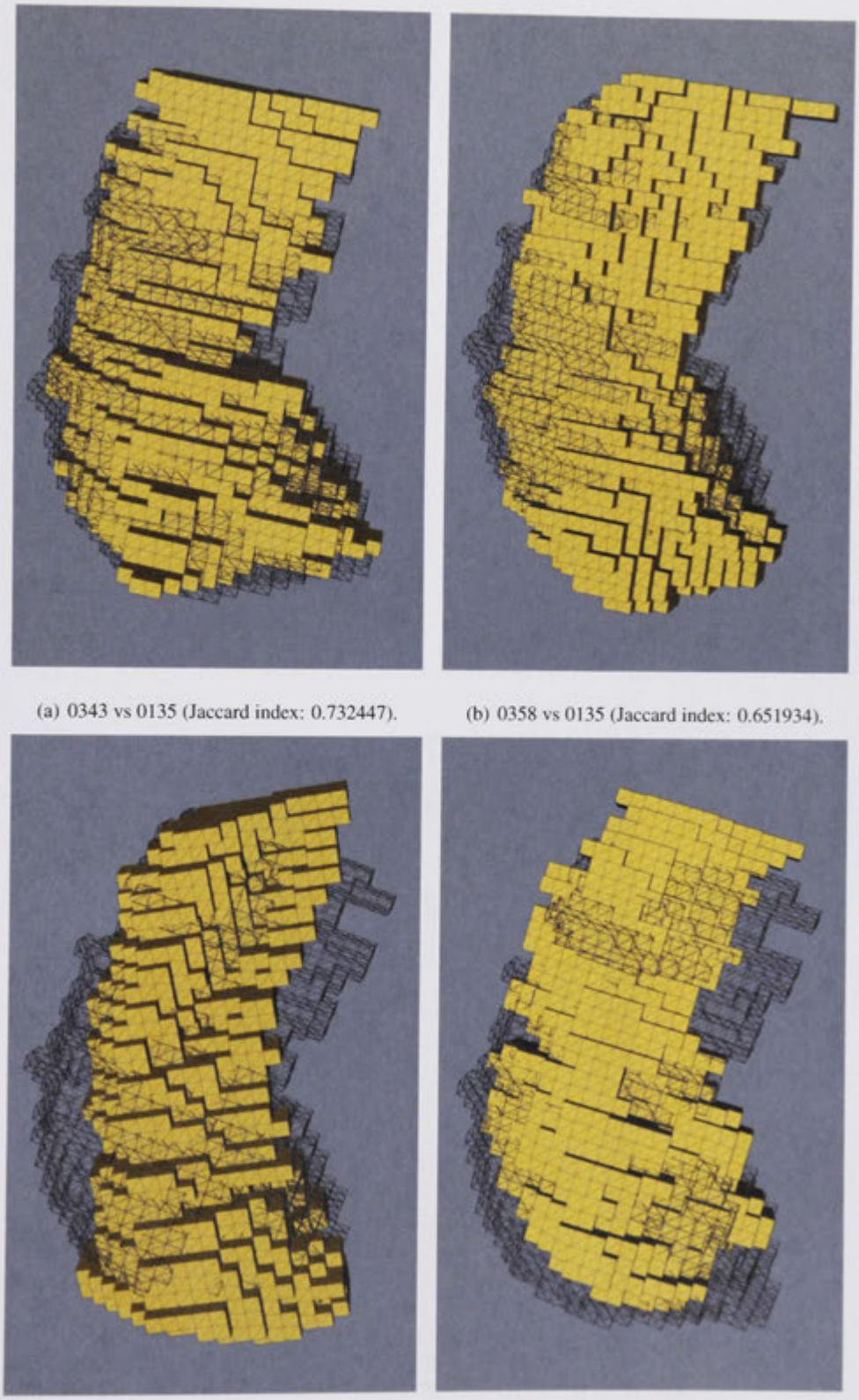
(a) 0292 vs 0135 (Jaccard index: 0.732912).

(b) 0293 vs 0135 (Jaccard index: 0.653399).

(c) 0303 vs 0135 (Jaccard index: 0.742436).

(d) 0322 vs 0135 (Jaccard index: 0.73724).

Figure 4: 3D rigid body registration of OASIS data set. The wireframe hippocampus is 0135.



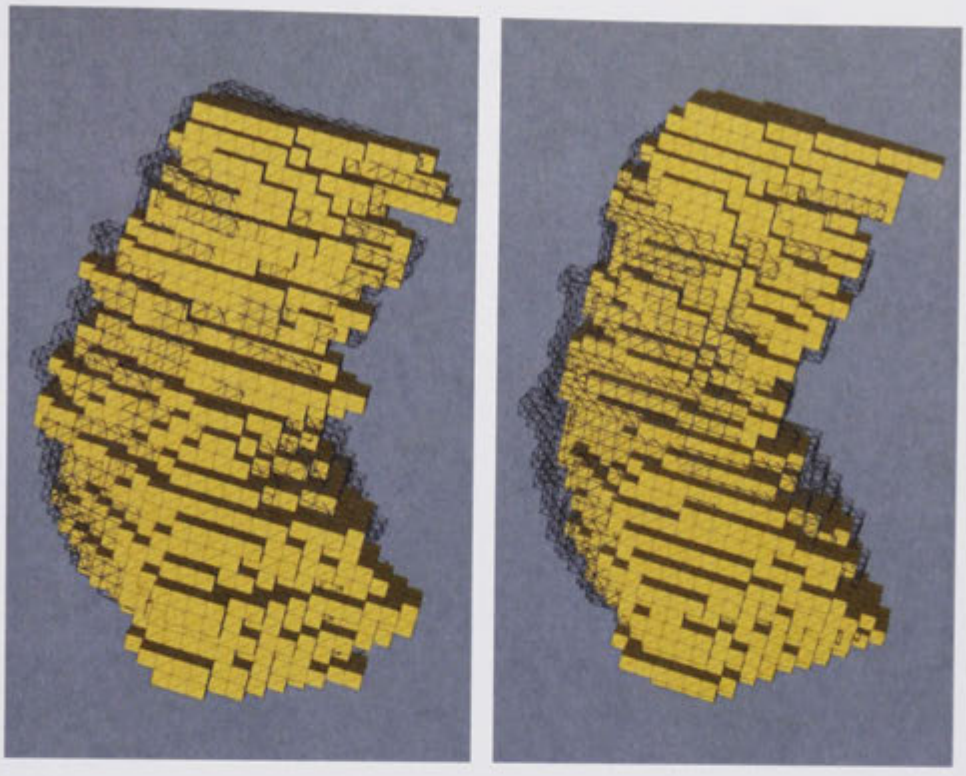
(a) 0343 vs 0135 (Jaccard index: 0.732447).

(b) 0358 vs 0135 (Jaccard index: 0.651934).

(c) 0422 vs 0135 (Jaccard index: 0.610259).

(d) 0438 vs 0135 (Jaccard index: 0.704253).

Figure 5: 3D rigid body registration of OASIS data set. The wireframe hippocampus is 0135.



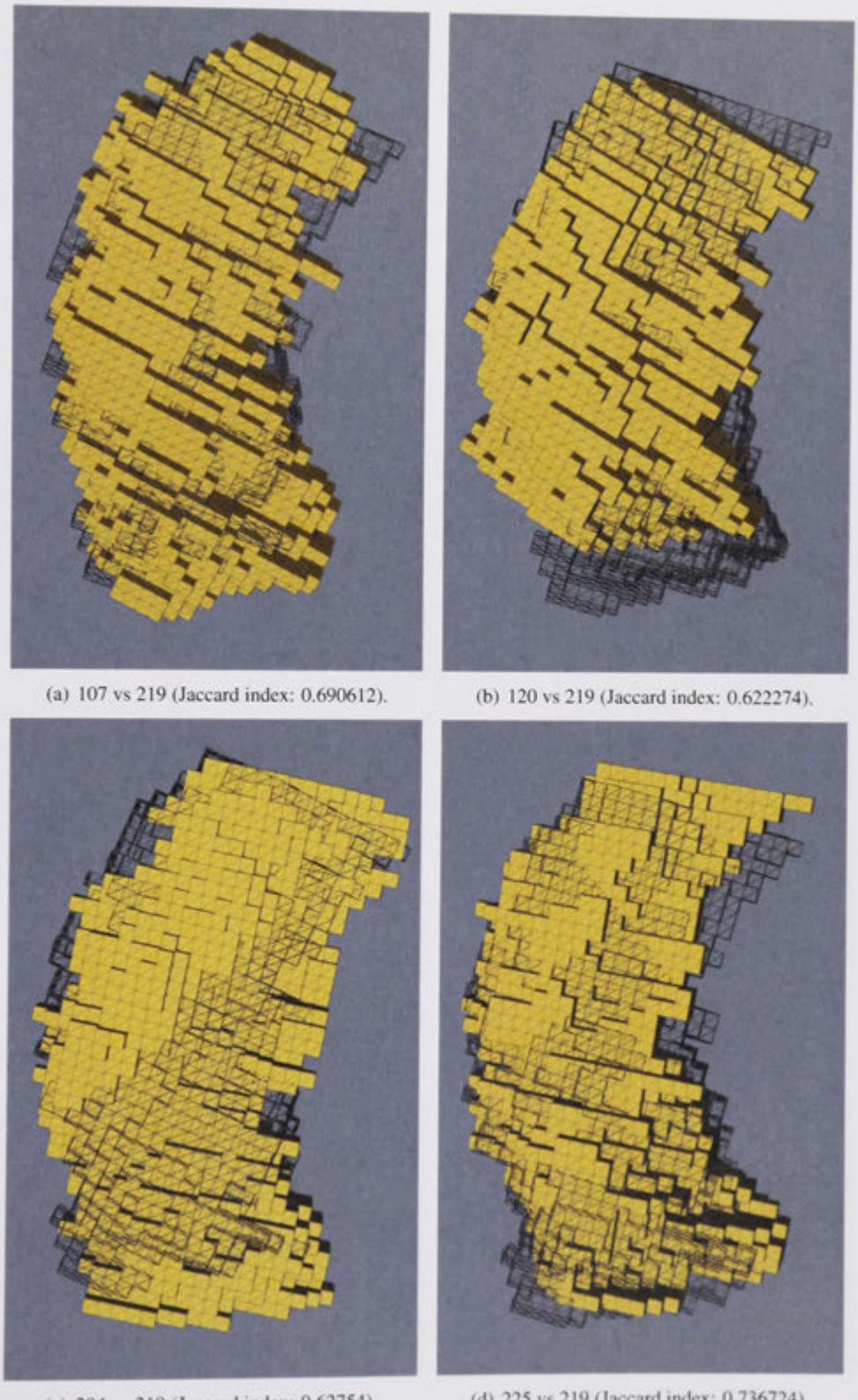
(a) 0455 vs 0135 (Jaccard index: 0.775579).

(b) 0456 vs 0135 (Jaccard index: 0.715522).

(c) 0457 vs 0135 (Jaccard index: 0.625921).

Figure 6: 3D rigid body registration of OASIS data set. The wireframe hippocampus is 0135.

2. PATH



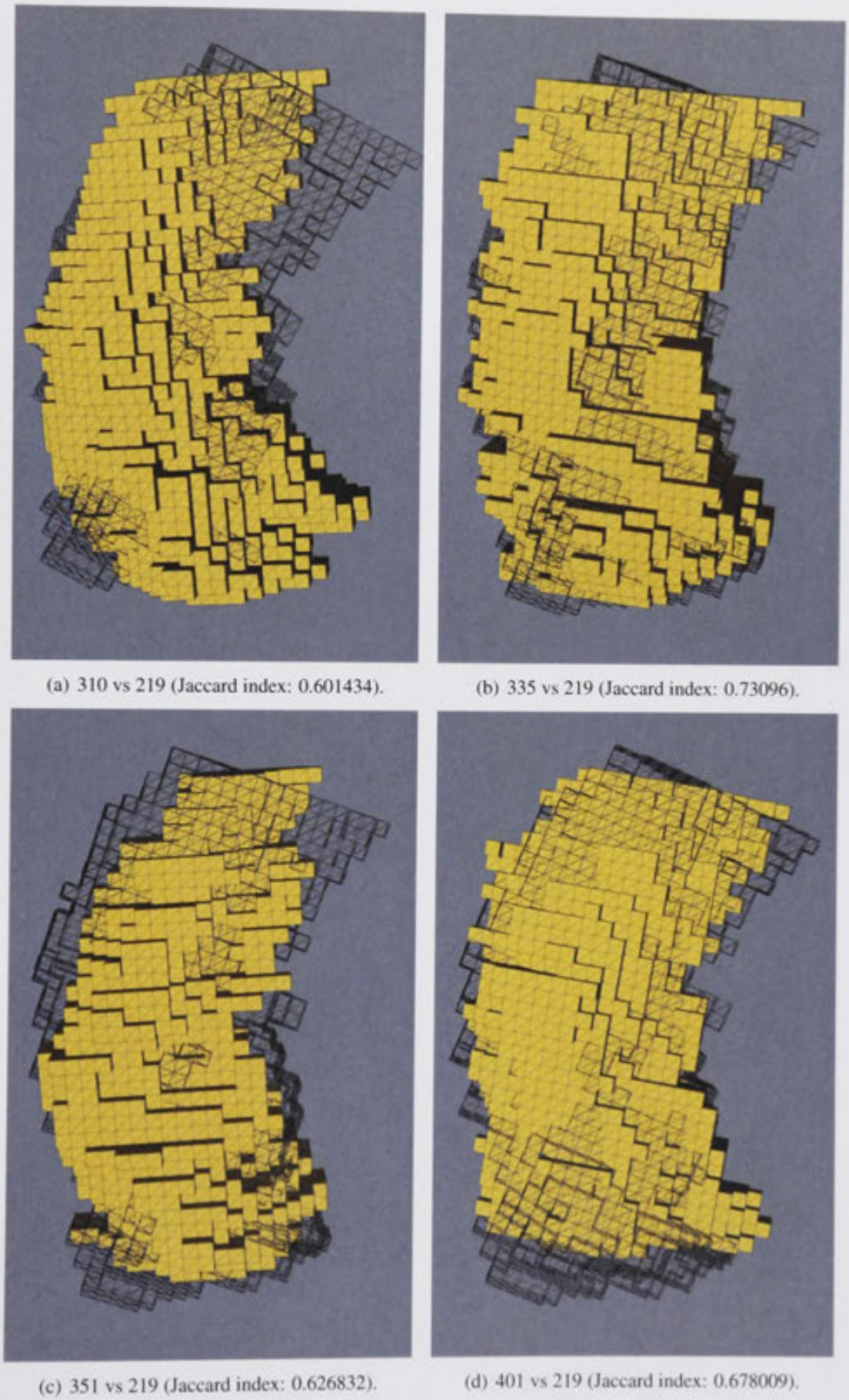
(a) 107 vs 219 (Jaccard index: 0.690612).

(b) 120 vs 219 (Jaccard index: 0.622274).

(c) 204 vs 219 (Jaccard index: 0.62754).

(d) 225 vs 219 (Jaccard index: 0.736724).

Figure 7: 3D rigid body registration of PATH data set. The wireframe hippocampus is 219.



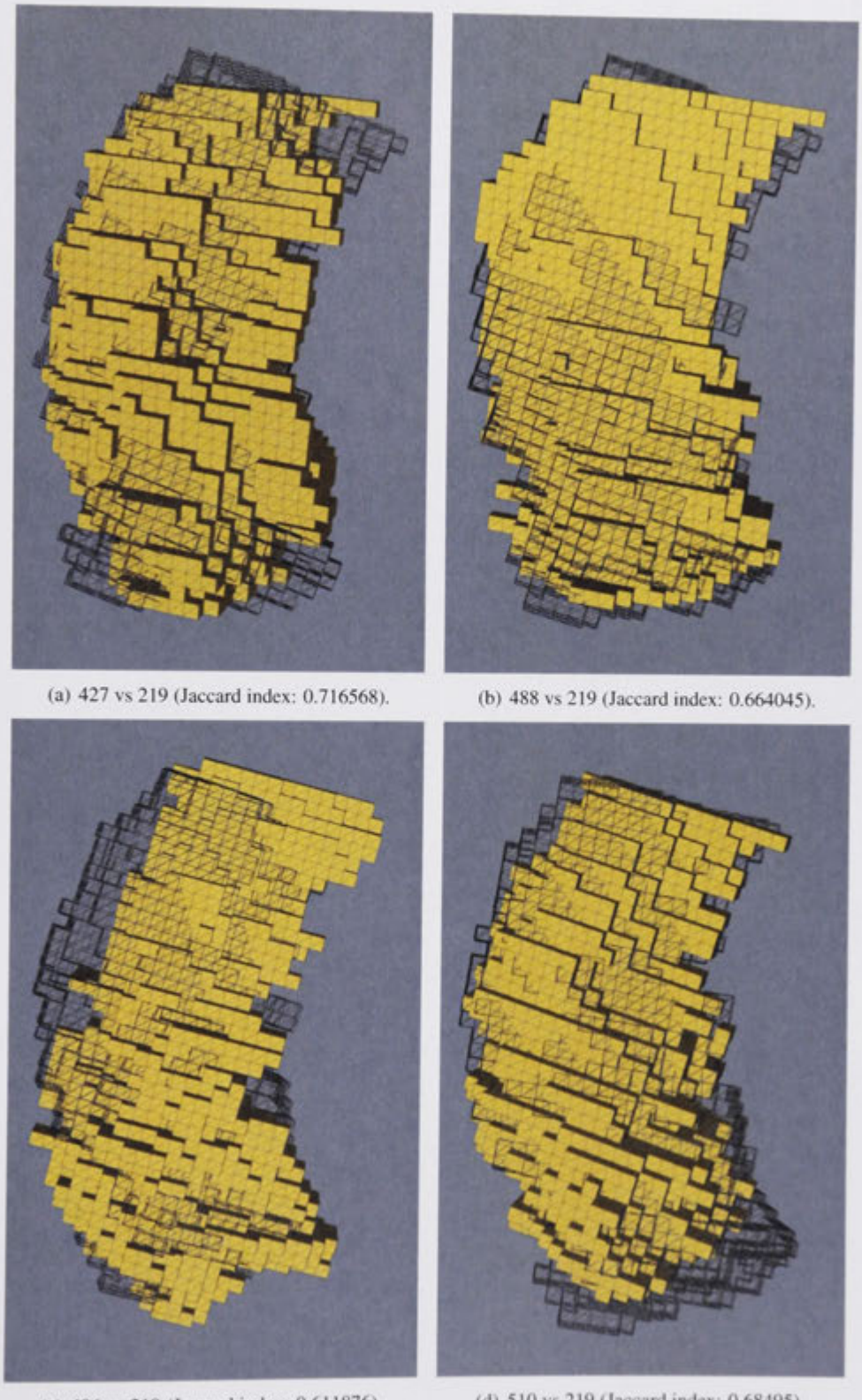
(a) 310 vs 219 (Jaccard index: 0.601434).

(b) 335 vs 219 (Jaccard index: 0.73096).

(c) 351 vs 219 (Jaccard index: 0.626832).

(d) 401 vs 219 (Jaccard index: 0.678009).

Figure 8: 3D rigid body registration of PATH data set. The wireframe hippocampus is 219.



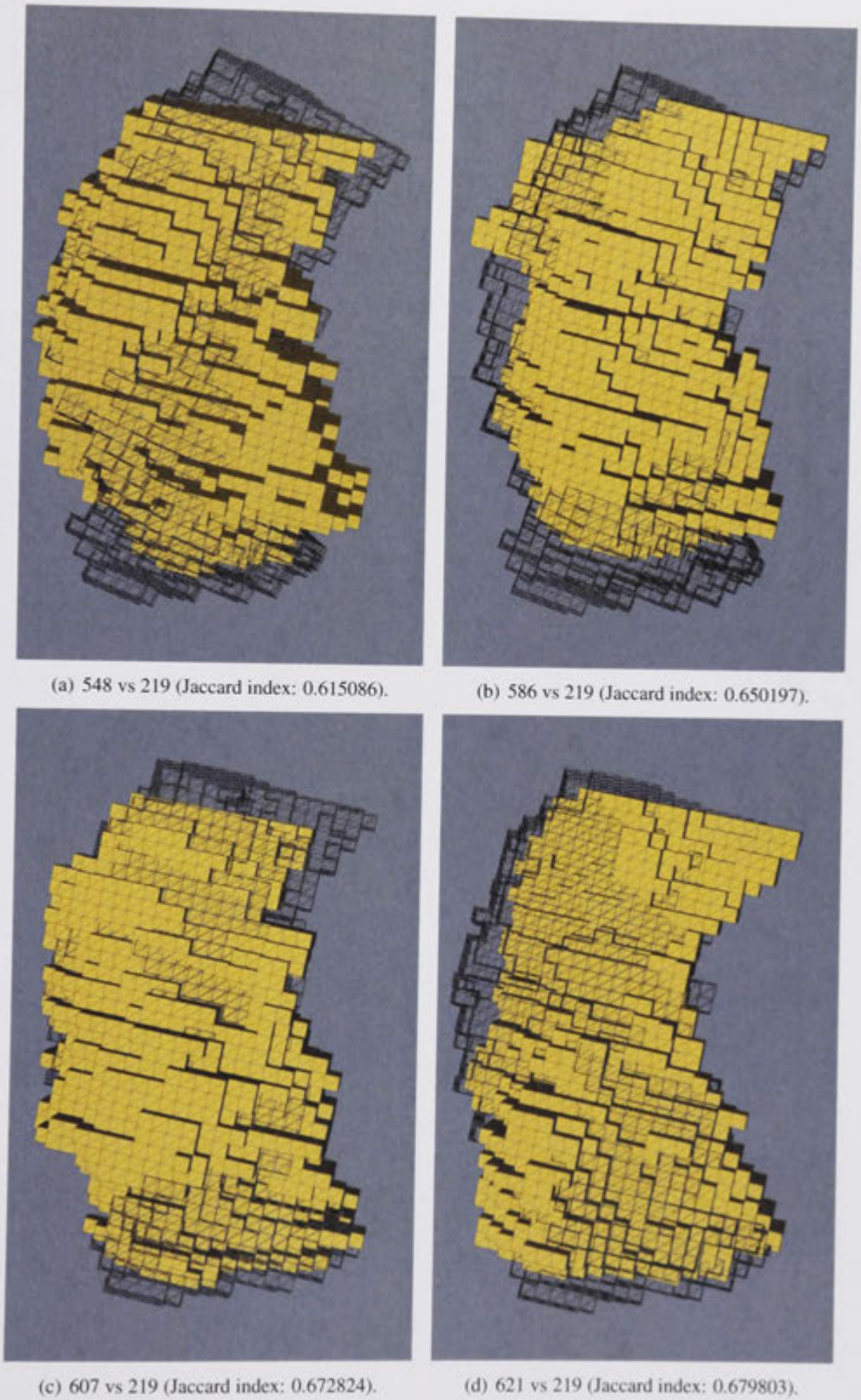
(a) 427 vs 219 (Jaccard index: 0.716568).

(b) 488 vs 219 (Jaccard index: 0.664045).

(c) 496 vs 219 (Jaccard index: 0.611876).

(d) 510 vs 219 (Jaccard index: 0.68495).

Figure 9: 3D rigid body registration of PATH data set. The wireframe hippocampus is 219.



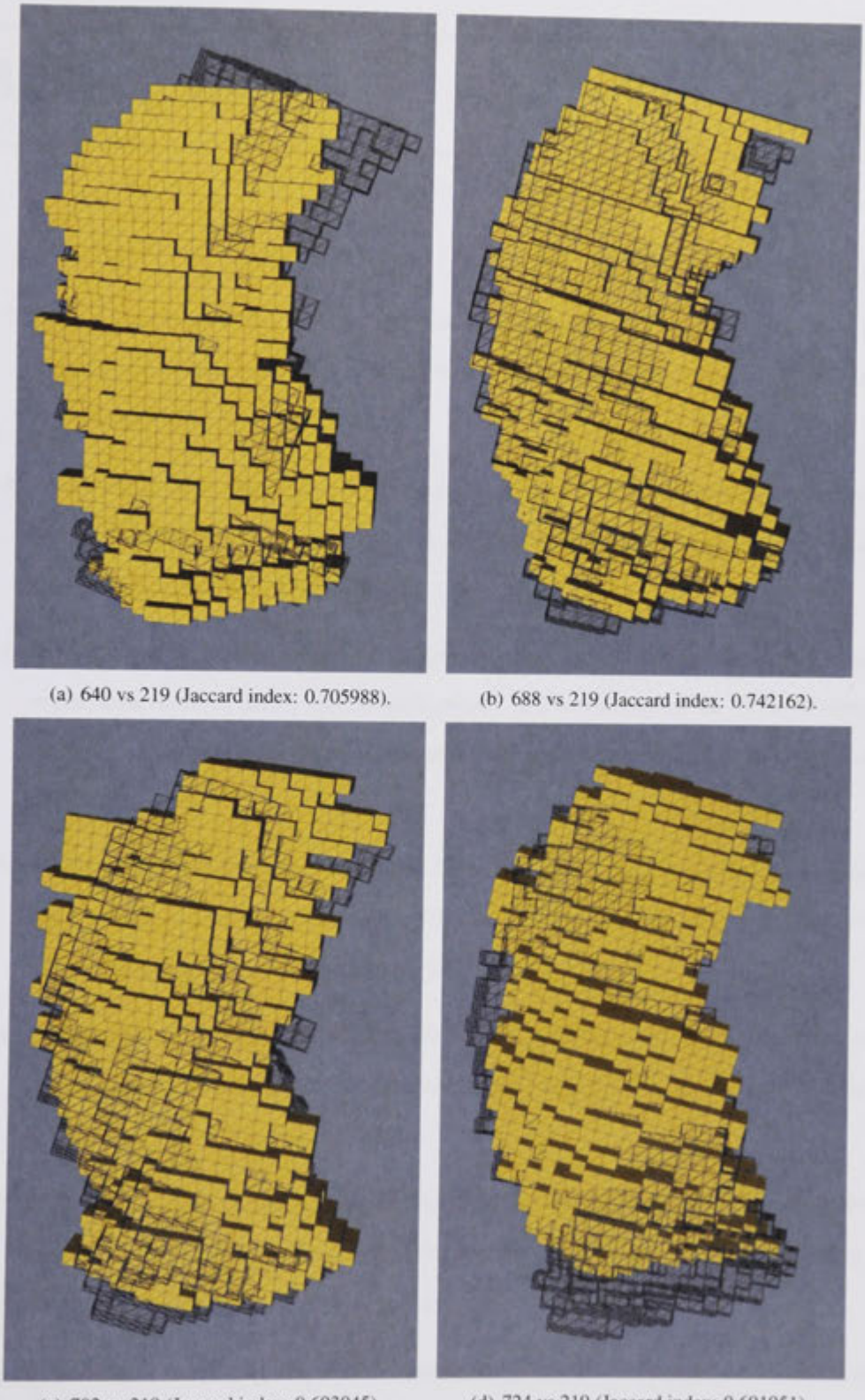
(a) 548 vs 219 (Jaccard index: 0.615086).

(b) 586 vs 219 (Jaccard index: 0.650197).

(c) 607 vs 219 (Jaccard index: 0.672824).

(d) 621 vs 219 (Jaccard index: 0.679803).

Figure 10: 3D rigid body registration of PATH data set. The wireframe hippocampus is 219.



(a) 640 vs 219 (Jaccard index: 0.705988).

(b) 688 vs 219 (Jaccard index: 0.742162).

(c) 702 vs 219 (Jaccard index: 0.693045).

(d) 724 vs 219 (Jaccard index: 0.691951).

Figure 11: 3D rigid body registration of PATH data set. The wireframe hippocampus is 219.

Bibliography

1. What is alzheimer's disease? In `http://alzheimers.org.uk`, August 2007.

2. Log transformation. In `http://biopuce.insa-toulouse.fr/ExperimentExplorer/doc/BioPlot/node25.html`, Date of downloading: 08 May 2011.

3. A. Albarelli, S. R. Bulo, A. Torsello, and M. Pelillo. Matching as a non-cooperative game. In *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV'09)*, pages 1319–1326, 2009.

4. A. Albarelli, E. Rodola, and A. Torsello. A game-theoretic approach to fine surface registration without initial motion estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'10)*, pages 430–437, 2010.

5. A. Albarelli, E. Rodola, and A. Torsello. Robust game-theoretic inlier selection for bundle adjustment. In *Proceedings of the Fifth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'10)*, 2010.

6. D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Neural Information Processing Systems*, 2004.

7. V. Arsigny, X. Pennec, and N. Ayache. Polyrigid and polyaffine transformations: A novel geometrical tool to deal with non-rigid deformations – application to the registration of histological slices. *Medical Image Analysis*, 9(6):507–523, 2005.

8. K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

9. J. Ashburner. A fast diffeomorphic image registration algorithm. *NeuroImage*, 38(1):95–113, October 2007.

10. B. Avants and J. C. Gee. Geodesic estimation for large deformation anatomical shape averaging and interpolation. *NeuroImage*, 23(Supplement 1):S139–S150, 2004.

11. B. Avants and J. C. Gee. Shape averaging with diffeomorphic flows for atlas creation. In *IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pages 595–598, April 2004.

12. B. Avants and J. C. Gee. Symmetric geodesic shape averaging and shape interpolation. In M. Sonka, I. A. Kakadiaris, and J. Kybic, editors, *Mathematical Methods in Biomedical Image Analysis*, volume 3117 of *Lecture Notes in Computer Science*, pages 99–110, Heidelberg, 2004. Springer-Verlag.

13. R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46(1):1–21, 1989.

14. M. Basso, J. Yang, L. Warren, M. G. MacAvoy, P. Varma, R. A. Bronen, and C. H. van Dyck. Volumetry of amygdala and hippocampus and memory performance in alzheimer's disease. *Psychiatry Research: Neuroimaging*, 146(3):251–261, 2006.

15. M. F. Beg, M. I. Miller, A. Trouve, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157, February 2005.

16. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

17. J. Berkmann and T. Caelli. Computation of surface geometry and segmentation using covariance techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1114–1116, 1994.

18. G. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding: Turbo codes. In *Proceedings of the 1993 International Conference on Communications*, pages 1064–1070, 1993.

19. P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

20. A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

21. F. L. Bookstein. Thin-plate splines and the atlas problem for biomedical images. In *Proceedings of the 12th International Conference on Information Processing in Medical Imaging*, pages 326–342, 1991.

22. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1–18, 2001.

23. H. Braak and E. Braak. Evolution of neuronal changes in the course of alzheimer's disease. *Journal of Neural Transmission Supplementum*, 53:127–140, 1998.

24. C. Brechbuhler, G. Gerig, and O. Kubler. Parametrization of closed surfaces for 3-d shape description. *Computer Vision and Image Understanding*, 61(2):154–170, March 1995.

25. P. Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, New York, 1999.

26. R. Brookmeyer, E. Johnson, K. Ziegler-Graham, and H. M. Arrighi. Forecasting the global burden of alzheimer's disease. *Alzheimer's & Dementia*, 3(3):186–191, 2007.

27. I. J. Bruno, N. M. Kemp, P. J. Artymiuk, and P. Willett. Representation and searching of carbohydrate structures using graph-theoretic techniques. *Carbohydrate Research*, 304(1):61–67, October 1997.

28. T. Caelli, E. Osman, and G. West. 3d shape matching and inspection using geometric features and relational learning. *Computer Vision and Image Understanding*, 72(3):340–350, 1998.

29. T. M. Caelli and T. Caetano. Recent developments in the extraction and matching of image structure and syntax: From relaxation to junction tree models. In *Proceedings of PRASA 2003*, pages 1–8, 2003.

30. Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, April 1992.

31. G. C. H. Chuang and C. C. J. Kuo. Wavelet descriptor of planar curves: Theory and applications. *IEEE Transactions on Image Processing*, 5(1):56–70, 1996.

32. J. M. Coughlan and S. J. Ferreira. Finding deformable shapes using loopy belief propagation. In *Proceedings of the European Conference on Computer Vision (ECCV'02)*, pages 53–73, 2002.

33. J. Csernansky, L. Wang, J. Swank, J. Miller, M. Gado, D. McKeel, M. Miller, and J. Morris. Preclinical detection of alzheimer's disease: Hippocampal shape and volume predict dementia onset in the elderly. *Neuroimage*, 25(3):783–792, 2005.

34. H. Delingette, M. Hebert, and K. Ikeuchi. A spherical representation for the recognition of curved objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 103–112, 1993.

35. I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley & Sons Ltd, England, 1998.

36. G. Dudek and J. K. Tsotsos. Shape representation and recognition from multiscale curvature. *Computer Vision and Image Understanding*, 68(2):170–189, November 1997.

37. F.Cazals and M.Pouget. Estimating differential quantities using polynomial fitting of osculating jets. In *Proceedings of the Eurographics Symposium on Geometry Processing*, volume 3, pages 177–275, 2003.

38. P. F. Felzenszwalb and D. R. Huttenlocher. Efficient belief propagation for early vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, pages 1:261–268, 2004.

39. L. Ferrarini, W. M. Palm, H. Olofsen, M. A. van Buchem, J. H. Reiber, and F. Admiraal-Behloul. Shape differences of the brain ventricles in alzheimer's disease. *NeuroImage*, 32(3):1060–1069, 2006.

40. M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

41. M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

42. A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13-14):1145–1153, December 2003.

43. T. Gatzke and C. Grimm. Tech report wucse-2004-9: Improved curvature estimation on triangular meshes. Technical report, Washington University in St. Louis, September 2003.

44. T. Gatzke, C. Grimm, M. Garland, and S. Zelinka. Curvature maps for local shape comparison. In *Proceedings of the IEEE Conference on Shape Modeling and Applications*, pages 244–253, June 2005.

45. S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

46. G. Gerig, M. Styner, D. Jones, D. Weinberger, and J. Lieberman. Shape analysis of brain ventricles using spharm. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA'01)*, pages 171–178, 2001.

47. G. Gerig, M. Styner, M. E. Shenton, and J. A. Lieberman. Shape versus size: Improved understanding of the morphology of brain structures. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'01)*, pages 24–32, 2001.

48. D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

49. M. Golebiowski, M. Barcikowska, and A. Pfeffer. Magnetic resonance imaging-based hippocampal volumetry in patients with dementia of the alzheimer type. *Dementia and Geriatric Cognitive Disorders*, 10(4):284–288, 1999.

50. G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1996.

51. D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51(2):271–279, 1989.

52. E. M. Haacke, R. W. Brown, M. R. Thompson, and R. Venkatesan. *Magnetic Resonance Imaging: Physical Principles and Sequence Design*. John Wiley & Sons, Inc., New York, 1999.

53. E. Haber and J. Modersitzki. A multilevel method for image registration. *SIAM Journal on Scientific Computing*, 27(5):1594–1607, 2006.

54. D. Hahnel, S. Thrun, and W. Burgard. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 915–920, 2003.

55. B. Hamann. Curvature approximation for triangulated surfaces. In *Springer Computing Supplementum*, pages 139–153, 1993.

56. P. J. Harrison. The hippocampus in schizophrenia: a review of the neuropathological evidence and its pathophysiological implications. *Psychopharmacology*, 174(1):151–162, 2004.

57. M. Hernandez, S. Olmos, and X. Pennec. Comparing algorithms for diffeomorphic registration: Stationary lddmm and diffeomorphic demons. In *Proceedings of the International Workshop on the Mathematical Foundations of Computational Anatomy (MFCA'08)*, 2008.

58. D. L. G. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes. Medical image registration. *Physics in Medicine and Biology*, 46(3):R1–R45, March 2001.

59. J. H. Holland. *Adaptation in Natural and Artificial Systems*. University Of Michigan Press, 1975.

60. B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1987.

61. B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.

62. B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.

63. R. A. Hummel and S. W. Zucker. On the foundations of relaxation labeling process. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(3):267–286, 1983.

64. N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: State-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509–530, April 2005.

65. A. E. Johnson and M. Hebert. Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16:635–651, 1998.

66. A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, May 1999.

67. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

68. G. Kastenmuller, H.-P. Kriegel, and T. Seidl. Similarity search in 3d protein databases. In *Proceedings of the German Conference on Bioinformatics (GCB'98)*, pages 1–3, 1998.

69. M. Kazhdan and T. Funkhouser. Harmonic 3d shape matching. In *International Conference on Computer Graphics and Interactive Techniques*, pages 191–191, 2002.

70. M. Kazhdan, T. Funkhouser, and S. Rusingkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the Eurographics Symposium on Geometry Processing*, pages 156–164, 2003.

71. A. Kelemen, G. Szekely, and G. Gerig. Elastic model-based segmentation of 3-d neuroradiological data sets. *IEEE Transactions on Medical Imaging*, 18(10):828–839, 1999.

72. R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. American Mathematical Society, Providence, Rhode Island, 1980.

73. J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

74. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

75. M. Kortgen, G. J. Park, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. In *The 7th Central European Seminar on Computer Graphics*, 2003.

76. H.-P. Kriegel, S. Brecheisen, P. Kroger, M. Pfeifle, and M. Schubert. Using sets of feature vectors for similarity search on voxelized cad objects. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 587–598, 2003.

77. F. R. Kschischang, B. J. Frey, and H. A. Leoliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

78. S. Lee, G. Wolberg, and S. Y. Shin. Scattered data interpolation with multilevel b-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244, 1997.

79. S.-M. Lee, A. L. Abbott, N. A. Clark, and P. A. Araman. A shape representation for planar curves by shape signature harmonic embedding. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1940–1947, 2006.

80. H. Lester, S. R. Arridge, K. M. Jansons, L. Lemieux, J. V. Hajnal, and A. Oatridge. Non-linear registration with the variable viscosity fluid algorithm. In *Proceedings of the 16th International Conference on Information Processing in Medical Imaging*, pages 238–251, 1999.

81. F. F. Leymarie. *Three-Dimensional Shape Representation via Shock Flows*. Ph.d. thesis, Brown University, 2003.

82. F. F. Leymarie and B. B. Kimia. The shock scaffold for representing 3d shape. In *Proceedings of the 4th International Workshop on Visual Form (IWVF'01)*, pages 216–229, 2001.

83. B. Li, S. A. Millington, D. D. Anderson, and S. T. Acton. Registration of surfaces to 3d images using rigid body surfaces. In *Fortieth Asilomar Conference on Signals, Systems and Computers, 2006. ACSSC'06.*, pages 416–420, 2006.

84. S. Z. Li. *Towards 3D Vision from Range Images: An Optimisation Framework and Parallel Distributed Networks*. Ph.d. thesis, University of Surrey, Guilford, Surrey, UK, 1991.

85. S. Z. Li. Markov random field models in computer vision. In *Proceedings of the European Conference on Computer Vision (ECCV'94)*, pages 361–370, 1994.

86. S. Z. Li. Invariant representation, matching and pose estimation of 3d space curves under similarity transformations. *Pattern Recognition*, 30(3):447–458, 1997.

87. S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, third edition, 2009.

88. Y. Li and E. R. Hancock. Face recognition using shading-based curvature attributes. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, pages 538–541, 2004.

89. P. Liang and J. S. Todhunter. Representation and recognition of surface shapes in range images: A differential geometry approach. *Computer Vision, Graphics, and Image Processing*, 52(10):78–109, 1990.

90. P. Lieby, N. Barnes, and B. McKay. Topological repair on voxel-based quadrangular meshes. In *MICCAI Workshop: Mathematical Foundations of Computational Anatomy: Geometrical and Statistical Methods for Modeling Biological Shape Variability*, 2006.

91. T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

92. Y. Liu and M. A. Rodrigues. Geometric understanding of rigid body transformation. In *IEEE International Conference on Robotics & Automation*, pages 1275–1280, 1999.

93. M. Locatelli. *Handbook of Global Optimization*, volume 2, chapter 6. Kluwer Academic Publishers, Dordrecht, Netherlands, 2002.

94. B. Luo and E. R. Hancock. Feature matching with procrustes alignment and graph editing. In *The Seventh International Conference on Image Processing And Its Applications*, pages 72–76, July 1999.

95. B. Luo and E. R. Hancock. Iterative procrustes alignment with the em algorithm. *Image and Vision Computing*, 20(5-6):377–396, 2002.

96. D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner. Open access series of imaging studies (oasis): Cross-sectional mri data in young, middle aged, nondemented, and demented older adults. *Journal of Cognitive Neuroscience*, 19(9):1498–1507, 2007.

97. D. Mateus, F. Cuzzolin, R. P. Horaud, and E. Boyer. Articulated shape matching by robust alignment of embedded representations. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV'07)*, pages 1–8, 2007.

98. D. Mateus, F. Cuzzolin, R. P. Horaud, and E. Boyer. Articulated shape matching using locally linear embedding and orthogonal alignment. In *IEEE Workshop on Non-rigid Registration and Tracking through Learning (NRTL'07)*, 2007.

99. D. Mateus, R. P. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer. Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'08)*, pages 1–8, 2008.

100. R. J. McEliece, D. J. MacKay, and J. F. Cheng. Turbo decoding as an instance of pearl's "belief propagation" algorithm. *IEEE Transactions on Selected Areas in Communication*, 16(2):140–152, 1998.

101. M. Mellor and M. Brady. Phase mutual information as a similarity measure for registration. *Medical Image Analysis*, 9(4):330–343, 2005.

102. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

103. M. Miller, A. Trouve, and L. Younes. On the metrics and euler-lagrange equations of computational anatomy. *Annual Review of Biomedical Engineering*, 4(1):375–405, 2002.

104. M. I. Miller and L. Younes. Group actions, homeomorphisms, and matching: A general framework. *International Journal of Computer Vision*, 41(1-2):61–84, January 2001.

105. F. Mohanna and F. Mokhtarian. An efficient active contour model through curvature scale space filtering. *Multimedia Tools and Applications*, 21:225–242, 2003.

106. F. Mokhtarian, S. Abbasi, and J. Kittler. Indexing an image database by shape content using curvature scale space. In *Proceedings of the IEE Colloquium on Intelligent Image Databases*, pages 4/1–4/6, May 1996.

107. F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *BMVC*. British Machine Vision Association, 1996.

108. F. Mokhtarian and M. Bober. *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization*. Kluwer Academic Publishers, 2003.

109. F. Mokhtarian and A. K. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):34–43, 1986.

110. R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

111. R. M. Neal. Tech report crg-tr-93-1: Probabilistic inference using markov chain monte carlo methods. Technical report, University of Toronto, September 1993.

112. J. Nolte and J. B. Angevine. *The Human Brain in Photographs and Diagrams*. Mosby, Inc., St. Louis, Missouri, 2000.

113. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, October 2002.

114. J. Pearl. Probabilistic reasoning in intelligent systems: Network of plausible inference, 1988. San Mateo, California: Margan Kaufmann Publishers, INC.

115. M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1119, 1999.

116. E. Persoon and K. Fu. Shape discrimination using fourier descriptors. *IEEE Transactions on Systems, Man and Cybernetics*, 7(3):170–179, 1977.

117. C. Peterson and B. Soderberg. A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, 1(1):3–22, 1989.

118. S. M. Pizer, D. S. Fritsch, P. A. Yushkevich, V. E. Johnson, and E. L. Chaney. Segmentation, registration, and measurement of shape variation via image object shape. *IEEE Transactions on Medical Imaging*, 18(10):851–865, October 1999.

119. A. Qiu, M. Albert, L. Younes, and M. I. Miller. Time sequence diffeomorphic metric mapping and parallel transport track time-dependent shape changes. *NeuroImage*, 45(1):S51–S60, March 2009.

120. A. Qiu, L. Younes, M. I. Miller, and J. G. Csernansky. Parallel transport in diffeomorphisms distinguishes the time-dependent pattern of hippocampal surface deformation due to healthy aging and the dementia of the alzheimer's type. *NeuroImage*, 40(1):68–76, March 2008.

121. M. Quicken, C. Brechbuhler, J. Hug, H. Blattman, and G. Szekely. Parameterization of closed surfaces for parametric surface description. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, volume 1, pages 354–360, 2000.

122. E. Rodola, A. Albarelli, and A. Torsello. A game-theoretic approach to robust selection of multi-view point correspondence. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR'10)*, pages 57–60, 2010.

123. I. E. Rube, M. Ahmed, and M. Kamel. Wavelet approximation-based affine invariant shape representation functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):323–327, February 2006.

124. B. J. S. and X. Tan. Control of autonomous swarms using gibbs sampling. In *43rd IEEE Conference on Decision and Control*, 2004.

125. D. Saupe and D. V. Vranic. 3d model retrieval with spherical harmonics and moments. In *Proceedings of the 23rd DAGM Symposium*, pages 392–397, September 2001.

126. P. H. Schonemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, March 1966.

127. H.-Y. Shum, M. Hebert, and K. Ikeuchi. On 3d shape similarity. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 526–531, 1996.

128. K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. Hamilton-jacobi skeletons. *International Journal of Computer Vision*, 48(3):215–231, 2002.

129. K. Siddiqi and S. Pizer. *Medial Representations – Mathematics, Algorithms and Applications*. Springer, 2008.

130. K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.

131. M. Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 3. Publish or Perish, Inc, 2005.

132. J. Starck and A. Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV'07)*, pages 1–8, 2007.

133. M. Styner and G. Gerig. Three-dimensional medial shape representation incorporating object variability. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'02)*, pages 2:651–656, 2002.

134. J. Sun, N. N. Zheng, and H. Y. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.

135. H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proceedings of the 2003 International Conference on Shape Modeling and Application (SMI'03)*, pages 130–139, 2003.

136. J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3d shape retrieval methods. In *Proceedings Shape Modeling International*, pages 145–156, 2004.

137. M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'03)*, pages 2:900–906, 2003.

138. Q. M. Tieng and W. W. Boles. Recognition of 2d object contours using the wavelet transform zero-crossing representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):910–916, August 1997.

139. A. Torsello, S. R. Bulo, and M. Pelillo. Grouping with asymmetric affinities: A game-theoretic perspective. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 292–299, 2006.

140. A. Trouve. Diffeomorphisms groups and pattern matching in image analysis. *International Journal of Computer Vision*, 28(3):213–221, July 1998.

141. J. van der Poel, C. W. D. de Almeida, and L. V. Batista. A new multiscale, curvature-based shape representation technique for image retrieval based on dsp techniques. In *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems*, pages 373–378, 2005.

142. B. C. Venuri, J. Ye, Y. Chen, and C. M. Leonard. Image registration via level-set motion: Applications to atlas-based segmentation. *Medical Image Analysis*, 7(1):1–20, 2003.

143. B. Walsh. Markov chain monte carlo and gibbs sampling, 2004. Lecture Notes for Advanced Topics in Biological Statistics, University of Arizona.

144. C. Wang and S. Mahadevan. Manifold alignment using procrustes analysis. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1120–1127, July 2008.

145. L. Wang, J. P. Miller, M. H. Gado, D. W. McKeel, M. Rothermich, M. I. Miller, J. C. Morris, and J. G. Csernansky. Abnormalities of hippocampal surface structure in very mild dementia of the alzheimer type. *NeuroImage*, 30(1):52–60, 2006.

146. C. Watson, C. R. Jack, and F. Cendes. Volumetric magnetic resonance imaging clinical applications and contributions to the understanding of temporal lobe epilepsy. *Arch Neurol.*, 54(12):1521–1531, 1997.

147. Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12(1):1–41, 2000.

148. Y. Weiss and W. T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. Technical Report UCB/CSD-99-1046, University of California Berkeley, 1999.

149. Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2001.

150. Y. Wen. Using log-transform to avoid underflow problem in computing posterior probabilities. Unpublished manuscript, MIT, 2007.

151. T. J. Willmore. *Riemannian Geometry*. Oxford University Press, 1996.

152. G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Springer, New York, 2003.

153. J. Xuan, Y. Wang, M. T. Freedman, T. Adali, and P. Shields. Nonrigid medical image registration by finite-element deformable sheet-curve models. *International Journal of Biomedical Imaging*, pages 1–9, 2006.

154. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Bethe free energy, kikuchi approximations, and belief propagation algorithms. Technical Report TR-2001-16, Mitsubishi Electric Research Laboratories, 2001.

155. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Exploring Artificial Intelligence in the New Millennium*, pages 239–269, 2003.

156. C. Zahn and R. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, 21(3):269–281, March 1972.

157. C. Zhang and T. Chen. Efficient feature extraction for 2d/3d objects in mesh representation. In *Proceedings of the 2001 International Conference on Image Processing*, volume 3, pages 935–938, 2001.

158. C. Zhang and T. Chen. Indexing and retrieval of 3d models aided by active learning. In *Proceedings of the Ninth ACM International Conference on Multimedia*, pages 615–616, 2001.

159. D. Zhang and G. Lu. Content-based shape retrieval using different shape descriptor: A comparative study. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'01)*, pages 1139–1142, August 2001.

160. W. Zhang. *Multimodal Cardiovascular Image Analysis Using Phase Information*. Ph.d. thesis, Department of Engineering Science, University of Oxford, 2007.

161. L. Zhou, P. Lieby, N. Barnes, C. Reglade-Meslin, J. Walker, N. Cherbuin, and R. Hartley. Hippocampal shape analysis for alzheimer's disease using an efficient hypothesis test and regularized discriminative deformation. *Hippocampus*, 19(6):533–540, 2009.